시간 적응 기법 개선 연구

A Study on Improvement of Time Adaptation Method

경태윤, 이재상, 허진영, 김규홍

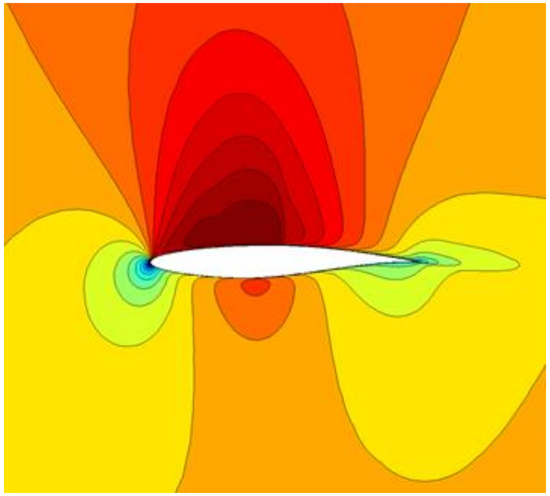
2021.07.08.

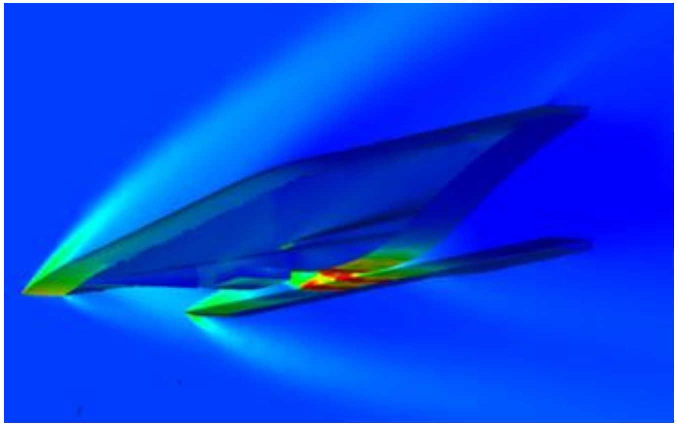
목 차

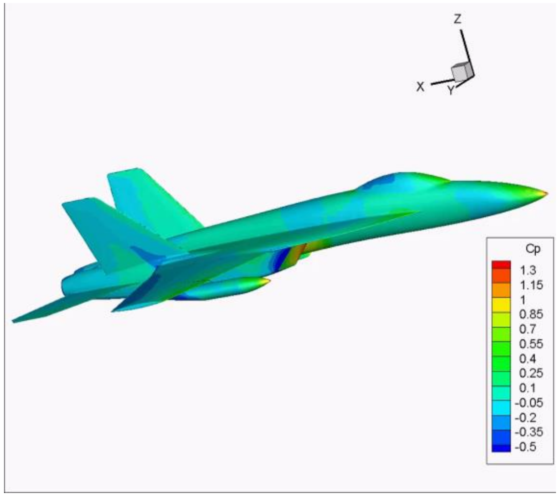
목 차

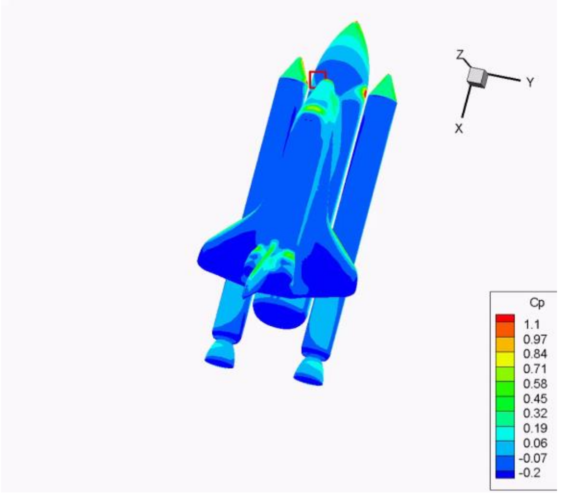
연구 개요

◆ CFD의 발전






Z
X
Y
Cp
1.3
1.15
1
0.85
0.7
0.55
0.4
0.25
0.1
-0.05
-0.2
-0.35
-0.5

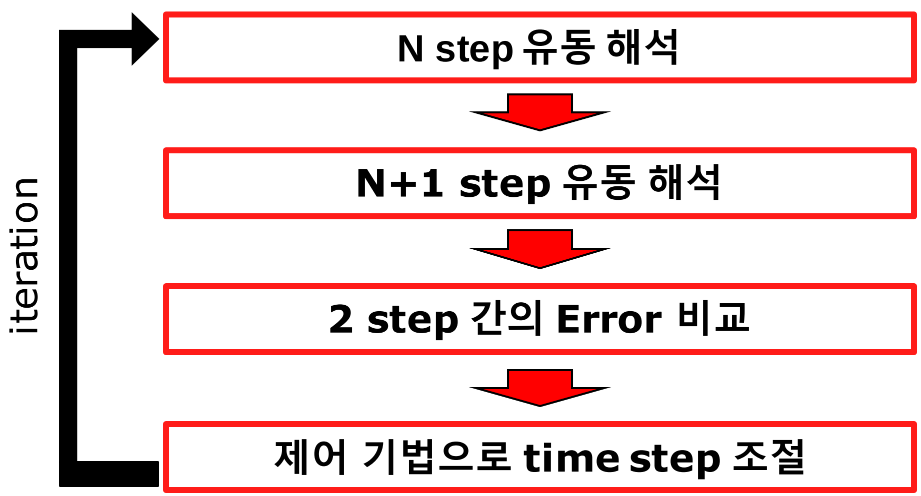

Z
Y
X
Cp
1.1
0.97
0.84
0.71
0.58
0.45
0.32
0.19
0.06
-0.07
-0.2

연구 개요

Time Adaptation

- 비정상 유동 해석은 적절한 timestep 선정이 매우 중요
  - Timestep 증가 -> 효율 증가 & 정확도 감소
  - Timestep 감소 -> 효율 감소 & 정확도 향상 – Trade-off

- 유동 해석의 정확도와 효율성을 높이기 위해 자동적으로 timestep 조절
  - Error의 경향을 분석하여 timestep을 자동적으로 조절
    - ▶ 유동 특성/변화 정도에 따른 timestep 변경
    - ▶ user의 timestep 선정 부담 감소

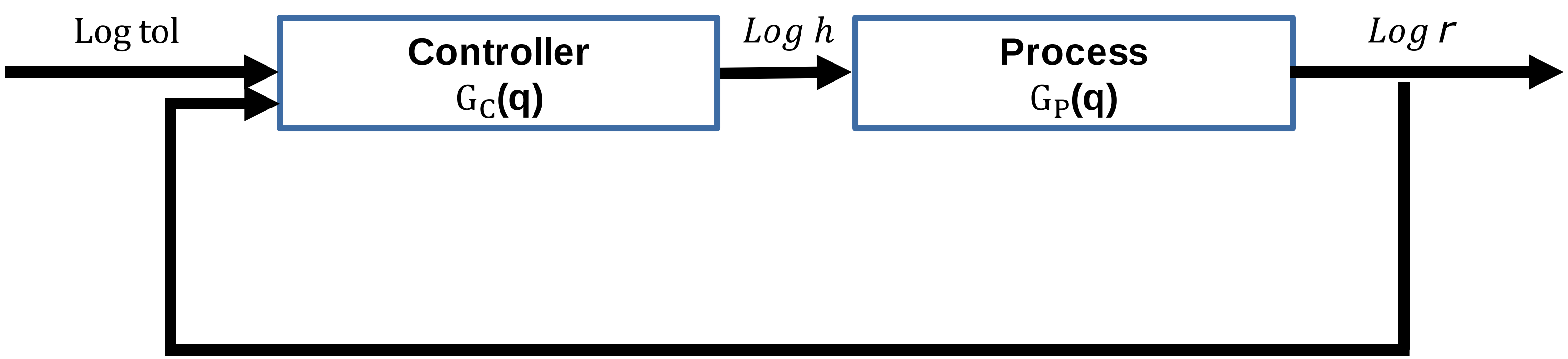


목 차

2. 시간 적응 기법

시간 적응 기법

- **Step Size Controller**
  - **Standard Controller**
    - Tolerance와 error r_n 비율을 이용하여 stepsize h_n 조정
      - $h_n = \gamma(\frac{tol}{r_n})^{\frac{1}{k}} h_{n-1}$
  - **PI Controller**
    - 기존 방법에 PI 제어 기법 적용
      - $h_n = (\frac{tol}{r_n})^{k_I}(\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$



시간 적응 기법

Comparison of Step Size Controller

Standard Controller

- $h_n = \gamma(\frac{tol}{r_n})^{\frac{1}{k}} h_{n-1}$
  - $logh_n = logh_{n-1} + \frac{1}{k}\left(log(\gamma^k\, tol) - logr_n\right)$ ▶It can be interpreted as Integral controller with $k_I = \frac{1}{k}$

$$logh_n = logh_{n-1} + k_I\left(log(\gamma^k\, tol) - logr_n\right)$$

PI Controller

- $h_n = (\frac{tol}{r_n})^{k_I} (\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$
  - $logh_n = logh_{n-1}$
    $+k_I(log(tol) - logr_n)$
    $+k_p[(log(tol) - logr_n) - (log(tol) - logr_{n-1})]$

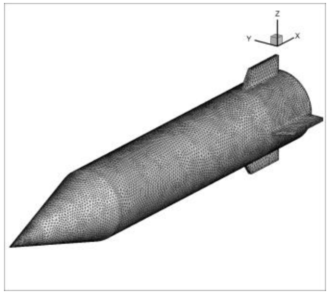
$$logh_n = logh_{n-1} + k_I(log(tol) - logr_n) + k_p[(log(tol) - logr_n) - (log(tol) - logr_{n-1})]$$

- Integral control means that h_n is changed such that offset between tol and r_n is zero.
- Proportional control works so that the difference between tol and r_n becomes 0 when there is a change in the size of r.

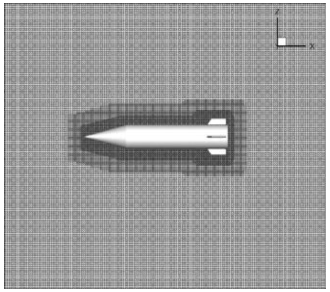
시간 적응 기법

이동 물체 해석 프로그램

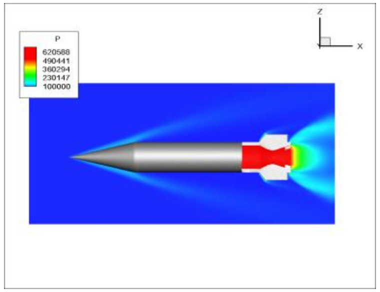
CAD 파일로 부터 표면 격자 자동 생성



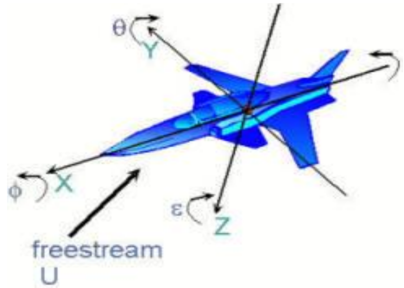
공간 질점 생성 및 Connectivity 구성



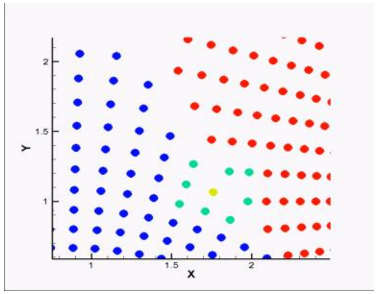
유동 해석 수행



6자유도 Dynamic Eq.으로 움직임 구현



질점 이동 및 Connectivity 재구성



목차

2

시간 적응 기법

- **LU-SGS에서의 적용**
- Runge-Kutta method에서의 적용

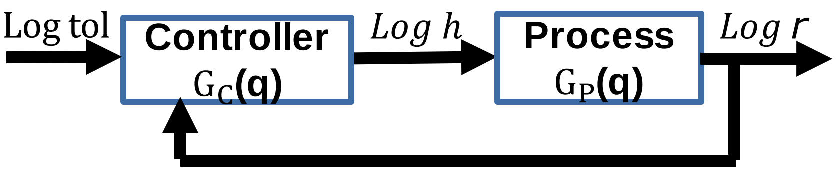
시간 적응 기법 설계

Step Size Control for LU-SGS

The Closed Loop Using PI Controller

$$h_n = \left(\frac{tol}{r_n}\right)^{k_I}\left(\frac{r_{n-1}}{r_n}\right)^{k_P} h_{n-1}$$

$k_I = 0.175, \qquad k_P = 0.075$



Error can be easily defined with any scalar value   Ex) ρ, p, T ...

$$r_n = \left(\frac{\int(\theta_n - \theta_{n-1})^2 d\Gamma}{\int \theta_n^2 d\Gamma}\right)^{\frac{1}{2}} \qquad \theta = \rho, p, T \ldots$$

- Application to Dual-time Stepping

$$\frac{\partial Q}{\partial t} = -R(Q)$$

$$\frac{\partial Q}{\partial \tau} = -\left[\frac{\partial Q}{\partial t} + R(Q)\right] = -R^*(Q)$$

$$\frac{\partial Q^{n+1}}{\partial \tau} = -\frac{1}{h_n}\left[-\frac{pp+2}{pp+1}Q^{n+1} + \frac{pp+1}{pp}Q^n - \frac{1}{pp(pp+1)}Q^{n-1}\right] - R(Q^{n+1}) = R^*(Q), \qquad pp = \frac{h_{n-1}}{h_n}$$

시간 적응 기법 설계

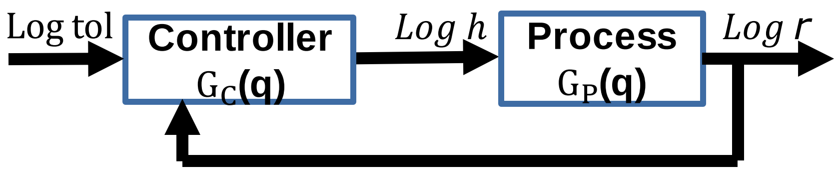
Step Size Control for LU-SGS

Order of LU-SGS & Dual time stepping

$$\frac{\partial Q}{\partial t} = -R(Q)$$

$$\frac{\partial Q}{\partial \tau} = -\left[\frac{\partial Q}{\partial t} + R(Q)\right] = -R^*(Q)$$

- Physical time discretization : 2nd order accurate
- Pseudo time discretization : 1st order accurate



Controller	Process
$h_n = (\frac{tol}{r_n})^{k_I}(\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$	$r_{n+1} = ch_n^k \quad (\mathrm{k} = 3)$
$log\ h_n = \left(k_I + k_P - \frac{k_p}{q}\right)[log\ tol - log\ r_n] + \frac{1}{q} log h_n$	$log\ r_{n+1} = klog\ h_n + logC$ $qlog\ r_n = klog\ h_n + logC$
$\frac{q-1}{q} log\ h_n = (k_I + k_P - \frac{k_p}{q})[log\ tol - log\ r_n]$	$log\ r_n = kq^{-1} log\ h_n + q^{-1} logC$
$\therefore G_c(q) = \frac{q(k_I + k_p) - k_p}{q-1}$	$\therefore G_p(q) = kq^{-1}$

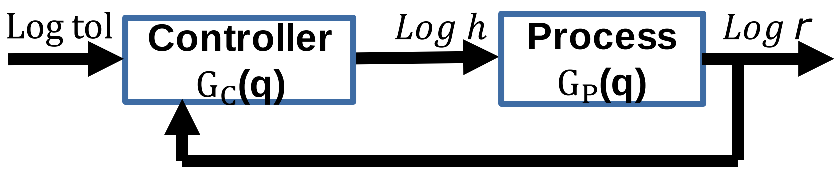
시간 적응 기법 설계

Step Size Control for LU-SGS

Order of LU-SGS & Dual time stepping

$$log\ r_n = G_p G_c(log\ tol - log\ r_n)$$

$$log\ r_n = \frac{G_p G_c}{1 + G_p G_c} log\ h_n$$



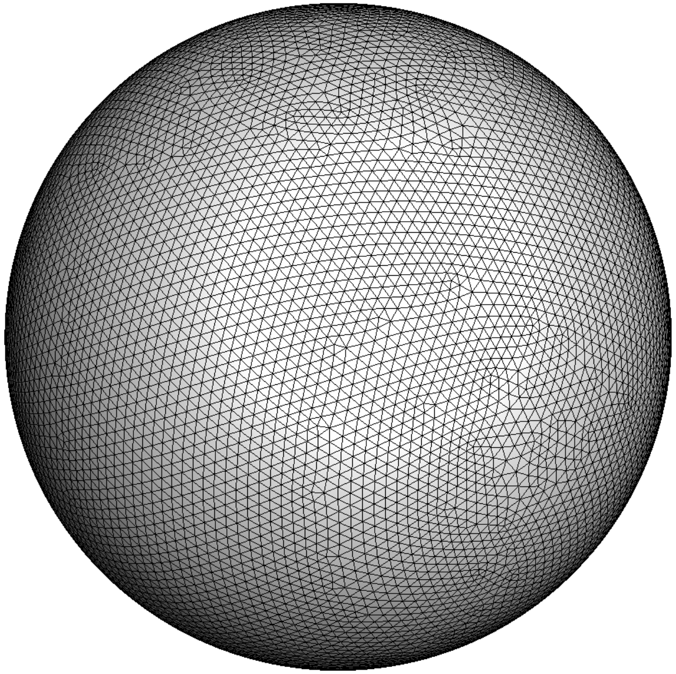
$$\frac{G_p G_c}{1 + G_p G_c} = \frac{(k_I + k_p)kq - kk_p}{q(q-1) + (k_I + k_p)kq - kk_p}$$

$$-1 \le \frac{1 - 3(k_I + k_p)}{2} \pm \sqrt{\frac{[1 - 3(k_I + k_p)]^2}{4} + 3k_p} \le 1$$

$$\therefore\ 0 < k_I < \frac{4}{3}, \qquad \frac{1}{3}\left(-3k_I + 2\sqrt{3}\sqrt{k_I} - 1\right) \le k_p \le \frac{1}{3} - \frac{k_I}{2}$$

Result. Fixed flow

- Time adaptation in fixed problem
  - 해석 방정식 : Euler Equation
  - 공간 차분 : M-AUSMPW+
  - 공간 내삽 : 3rd MLP
  - 시간 적분 : LU-SGS & Dual Time Stepping

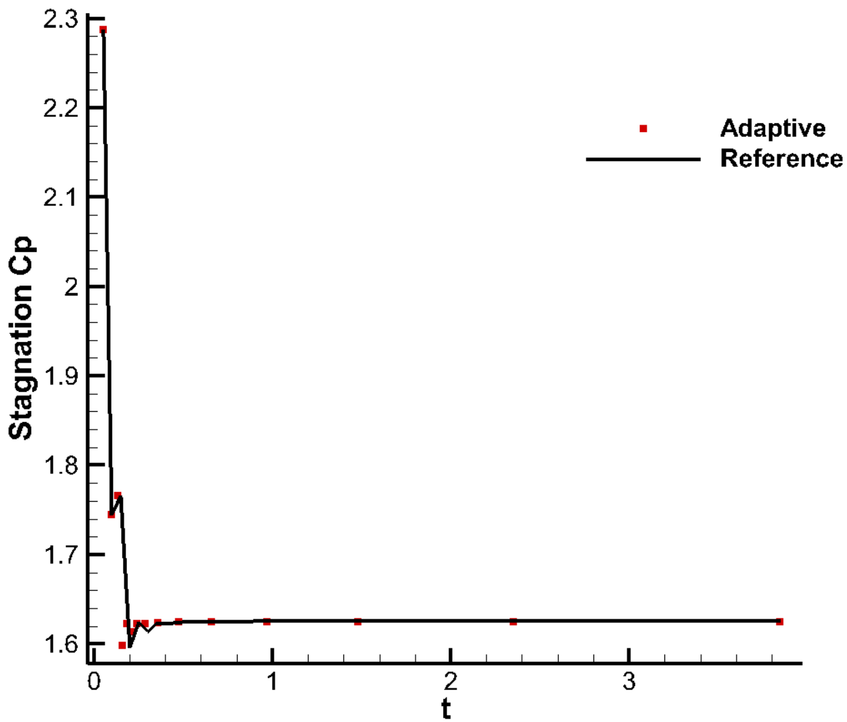


Operation condition	AOA (Deg)	자유류 속도 (Ma.)	온도(K)
	0.0	2.0	300.0

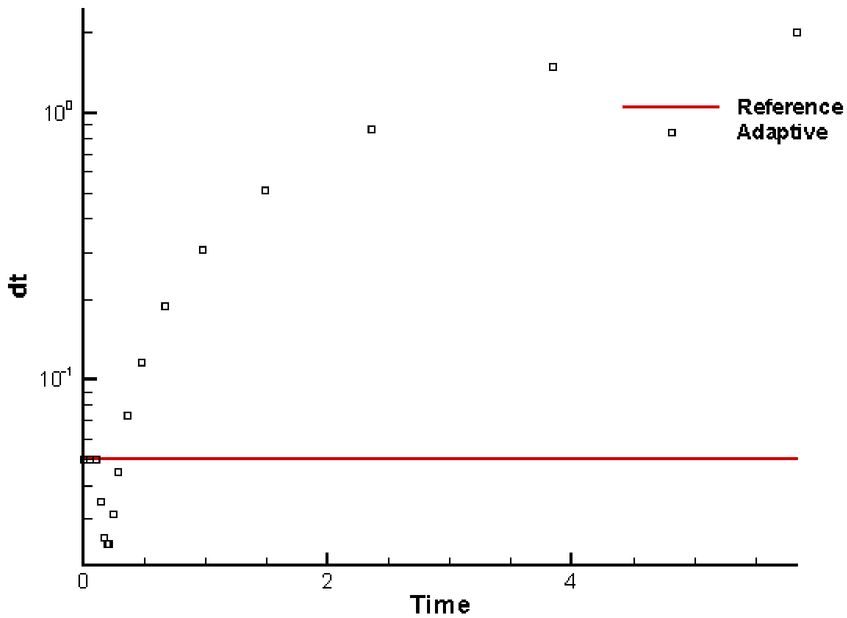
$$tolerance = 10^{-3} \,\&\, Starting\ DT = 0.05$$

Result. Fixed flow

◆ Time adaptation in fixed problem



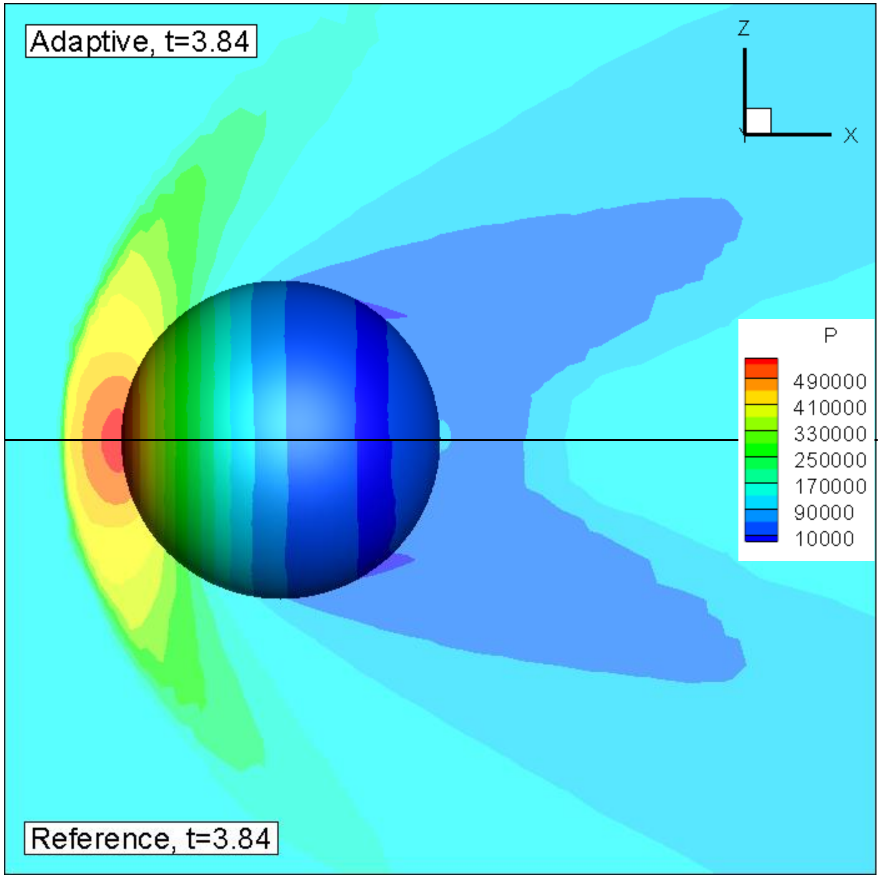
Stagnation Point Pressure Coefficient

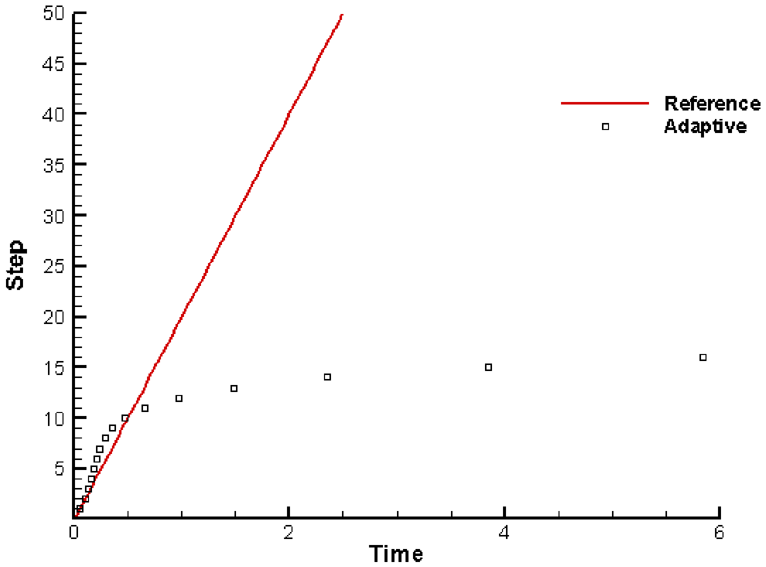


Physical Time Step Change

Result. Fixed flow

◆ Time adaptation in fixed problem





The Number of Calculated Timestep

16 Step .vs. 114 Step

- 7 times more efficient

시간 적응 기법 설계

- **Step Size Control for LU-SGS**
  - **Physical step size change & pseudo step size change**

N step

Step size h_n
Error r_n

Sub iteration

CFL → pseudo step size

Control CFL in each cell

$$\frac{\mathrm{CFL}_m}{\mathrm{CFL}_{m-1}} = \left(\frac{r_{m-2}}{r_{m-1}}\right)^{\frac{1}{k}}$$

Control h_n according to r_n

$$h_n = \left(\frac{tol}{r_n}\right)^{k_I}\left(\frac{r_{n-1}}{r_n}\right)^{k_P} h_{n-1}$$

N+1 step

Step size h_{n+1}
Error r_{n+1}

Sub iteration

CFL → pseudo step size

Control CFL in each cell

$$\frac{\mathrm{CFL}_m}{\mathrm{CFL}_{m-1}} = \left(\frac{r_{m-2}}{r_{m-1}}\right)^{\frac{1}{k}}$$

시간 적응 기법 설계

◆ Step Size Control for LU-SGS

Control Using I Controller

- **(1) Change CFL number by ratio of residual**

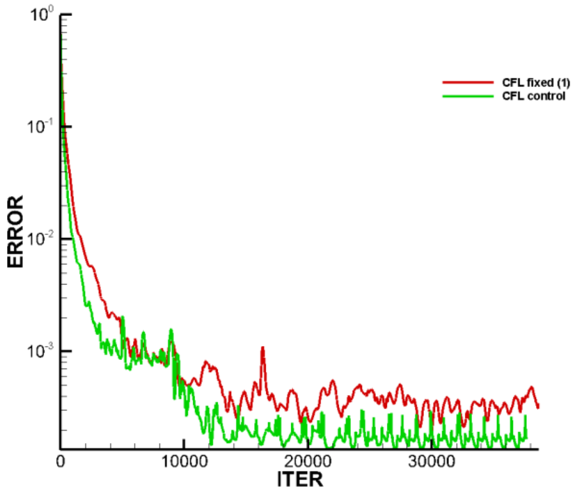
$$\mathrm{CFL}_m = \left(\frac{r_{m-2}}{r_{m-1}}\right)^{\frac{1}{k}} \mathrm{CFL}_{m-1}$$
$$r_m = |CONV(Q^m) - DIFF(Q^m)|$$

- **(2) Decrease CFL number until the solution change is bounded under maximum allowable change.**

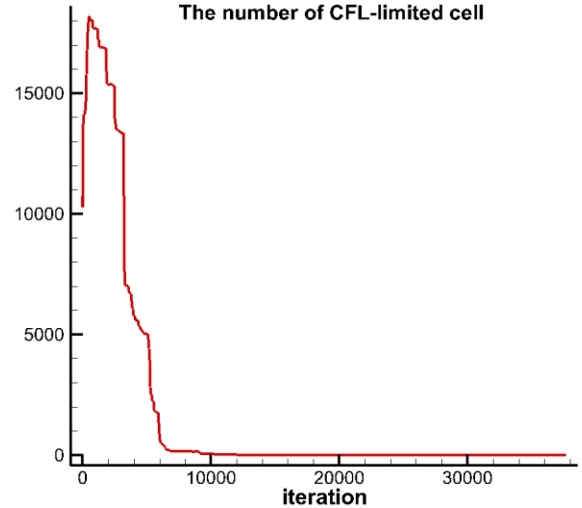
$$\Delta Q^{m+1} = \alpha\, Q^m$$

Result. CFL control

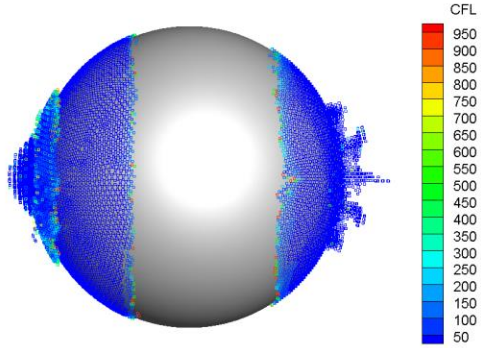
◆ **Sphere test**



Error history



The number of CFL-limited point



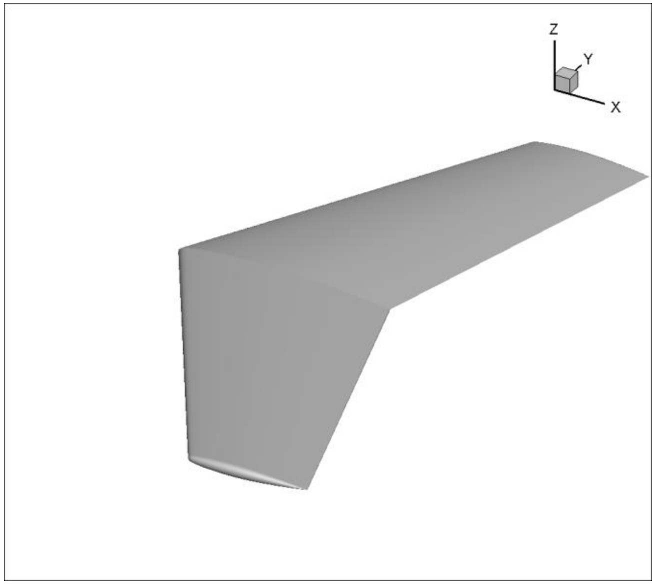
CFL-limited point
[iter=300]

Result. CFL control

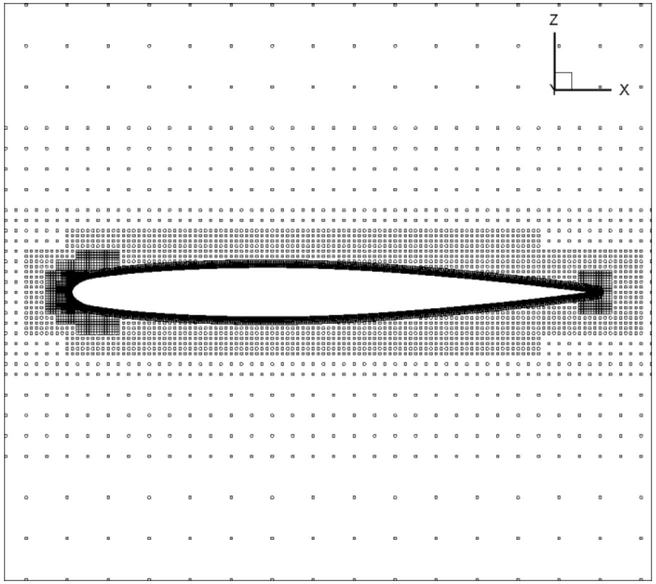
◆ **ONERA M6 Turbulence Analysis**

- Gov Eq. : Navier-Stokes Equation
- Space discretization : M-AUSMPW+ & 3^{rd} MLP
- Time integration : LU-SGS & Steady state

Operation condition	자유류 속도(Ma)	Re	받음각
	0.8395	11.4M	3.06



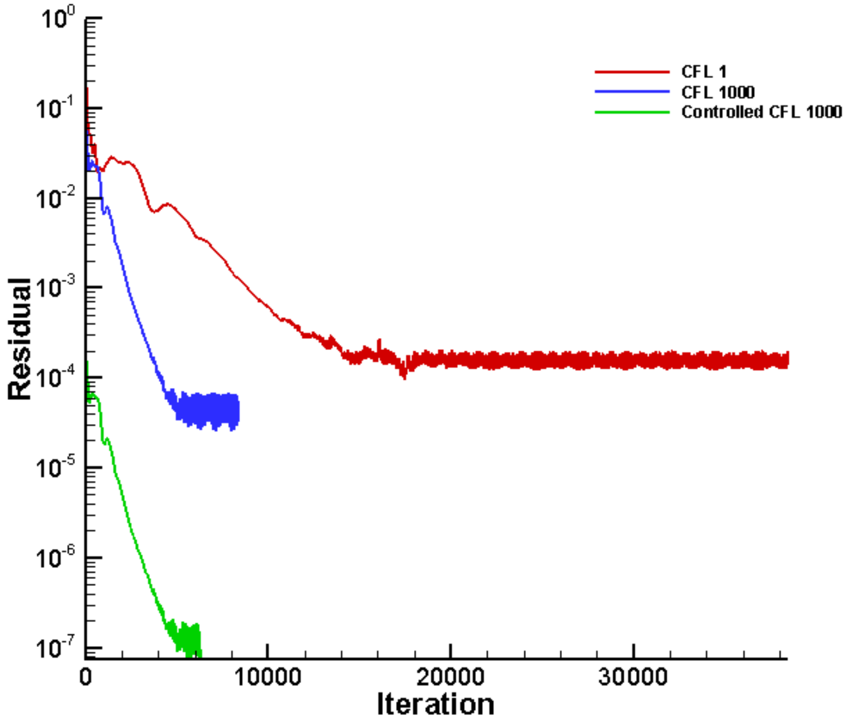
Onera M6 wing surface



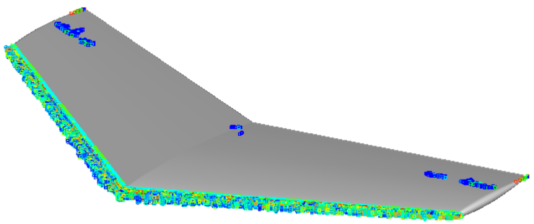
Point Distribution for Onera M6 wing

Result. CFL control

◆ **ONERA M6 test**



Residual history

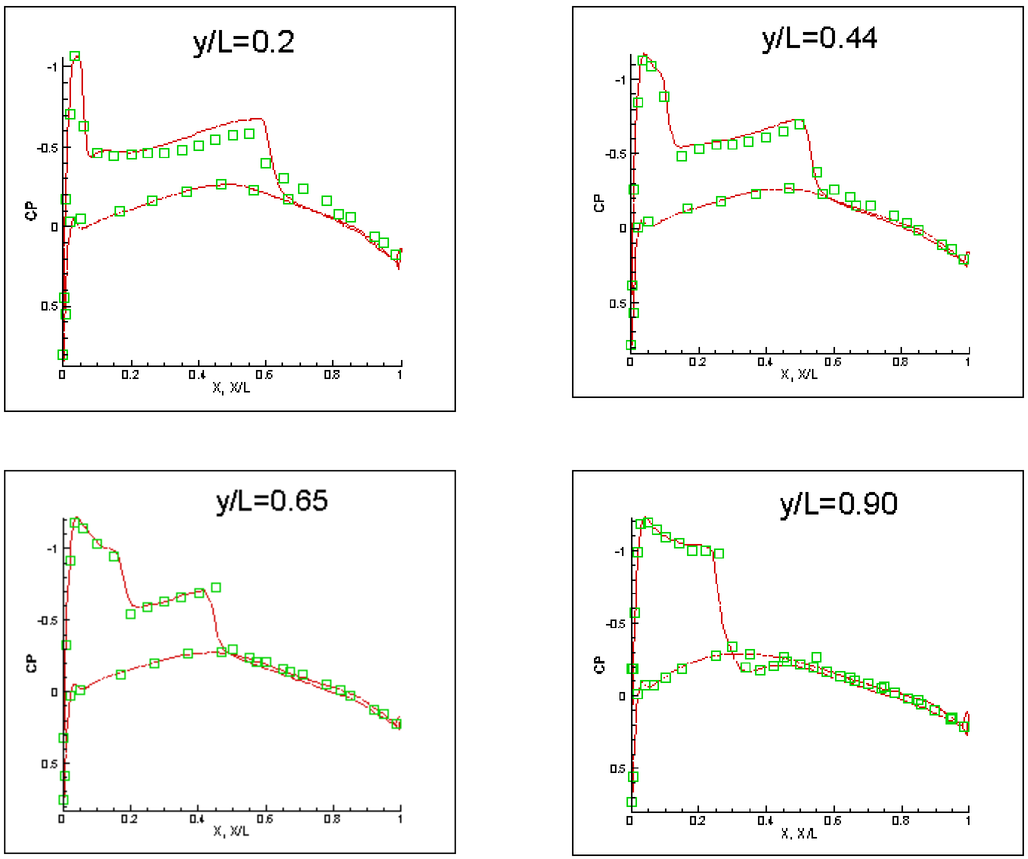


CFL-limited point
[iter=300]

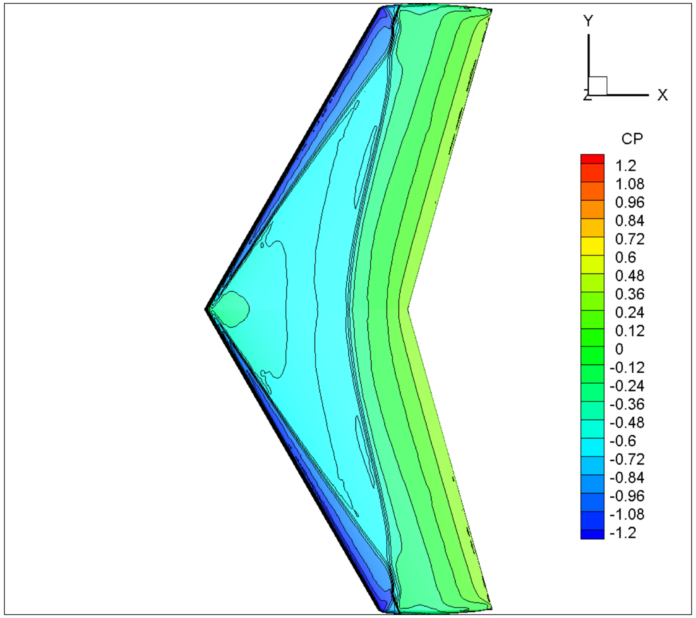
Result. CFL control

◆ ONERA M6 Turbulence Analysis

- **Surface pressure coefficient contour**
  - **표면 압력계수 분포 확인을 통한 충격파 위치와 강도 확인**



Pressure coefficient of Onera M6 wing



Pressure coefficient contour of Onera M6 wing

목차

2

시간 적응 기법

- LU-SGS에서의 적용
- **Runge-Kutta method에서의 적용**

시간 적응 기법 설계

◆ 6DOF Equation & Runge-Kutta 4th order method

Difficulties in time adaptation application

- Stability region is clearly limited in Runge-Kutta in contrast to LU-SGS
  - Rigorous selection of PI controller coefficient must be done
- No scalar value which can be used in error definition
  - Analysis for the equation itself must be done
- System equation

$$\begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{zx} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} \begin{bmatrix} \Sigma M_x + I_{yz}(Q^2 - R^2) + I_{zx}PQ - I_{xy}RP + \left(I_{yy} - I_{zz}\right)QR \\ \Sigma M_y + I_{zx}(R^2 - P^2) + I_{xy}QR - I_{yz}PQ + (I_{zz} - I_{xx})RP \\ \Sigma M_z + I_{xy}(P^2 - Q^2) + I_{yz}RP - I_{zx}QR + \left(I_{xx} - I_{yy}\right)PQ \end{bmatrix}$$

$$\begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} = \begin{bmatrix} VR - WQ \\ WP - UR \\ UQ - VP \end{bmatrix} + \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} + \begin{bmatrix} F_x/m \\ F_y/m \\ F_z/m \end{bmatrix}$$

$$\begin{bmatrix} \dot{q}_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

1.
Application to 1D differential equations

$$log\ r_n = \frac{G_p G_c}{1 + G_p G_c} log\ h_n$$

$$k_I = \frac{0.3}{k},\ k_P = \frac{0.4}{k}$$

(k is order or error**)**

2.
Error analysis of system differential equations

$$e_n = T\begin{bmatrix} c_{11} & & 0 \\ & \ddots & \\ 0 & & c_{mm} \end{bmatrix}\begin{bmatrix} \left\{\sum \frac{(h_1\rho_{1,1})^v}{v!}\right\} \cdots \left\{\sum \frac{(h_{n-1}\rho_{1,n-1})^v}{v!}\right\} \\ \vdots \\ \left\{\sum \frac{(h_1\rho_{m,1})^v}{v!}\right\} \cdots \left\{\sum \frac{(h_{n-1}\rho_{m,n-1})^v}{v!}\right\} \end{bmatrix}$$

The stability region of step size

$$\left|\sum \frac{(h_j\rho_{i,j})^v}{v!}\right| < 1$$

3.
Application to system differential equations

$$h_n = \left(\frac{tol}{r_n}\right)^{k_I}\left(\frac{r_{n-1}}{r_n}\right)^{k_P} h_{n-1}$$

(1) Control coefficient for 1D differential equation is still used.
(2) Error r_n is calculated by Richardson's extrapolation
(3) PI controller is switched to integral controller when any eigenvalue of system equations us outside the stability region.

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

M-stage k^{th} Runge-Kutta method

$$Y_1' = f(t_n, y_n), \qquad Y_i' = f\left(t_n + c_i h_n, y_n + h_n \sum a_{ij} Y_i'\right)$$

$$y_{n+1} = y_n + h_n \sum b_j Y_j', \qquad t_{n+1} = t_n + h_n$$

$$e_{n+1} = h_n \sum (b_j - \hat{b_j}) Y_j'$$

$$r_{n+1} = \begin{cases} \|e_{n+1}\| & error\ per\ step\ (EPS) \\ \dfrac{\|e_{n+1}\|}{h_n} & error\ per\ unit\ step\ (EPUS) \end{cases}$$

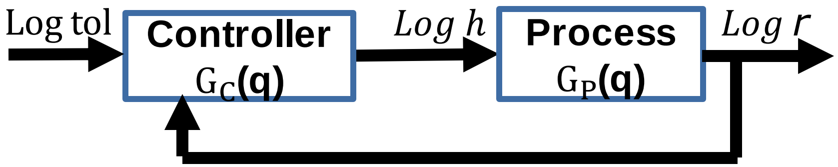
- **Error Analysis**
  - Provide transfer function for Process

If h_n is small,	If h_n is increased enough to approach stability region,
$y_{n+1} = P(h_n\lambda)y_n, e_{n+1} = E(h_n\lambda)y_n$ $r_{n+1} = \|\phi_n\|h_n^k, \quad \phi_n = y_n\lambda^{p_e}(\kappa_0 + \kappa_1 h_n\lambda + \cdots)$ $log\ r_n = G_{p1}(q) log\ h_n + q^{-1} log\|\phi_n\|$ $G_{p1}(q) = kq^{-1}$	$e_{n+1} = E(h_n\lambda)y_n = E(h_n\lambda)P(h_{n-1}\lambda)y_{n-1}$ $= E(h_n\lambda)\dfrac{P(h_{n-1}\lambda)}{E(h_{n-1}\lambda)}e_n$ $= P(h_s\lambda)\left(\dfrac{h_n}{h_s}\right)^{C_1}\left(\dfrac{h_{n-1}}{h_s}\right)^{-C_1+C_2} e_n$ $log\ r_n = G_{p2}(q)(log h_n - log h_s), G_{p2}(q) = \dfrac{C_1 q + C_2 - C_1}{q(q-1)}$

시간 적응 기법 설계

Step Size Control for Runge-Kutta

The Closed Loop Using Standard Controller



- $h_n = \gamma(\frac{tol}{r_n})^{\frac{1}{k}} h_{n-1}$

$log h_n = \frac{1}{k}\frac{q}{q-1}\left(log\left(\gamma^k\ tol\right) - log r_n\right)$ ▶It can be interpreted as Integral controller with $k_I = \frac{1}{k}$

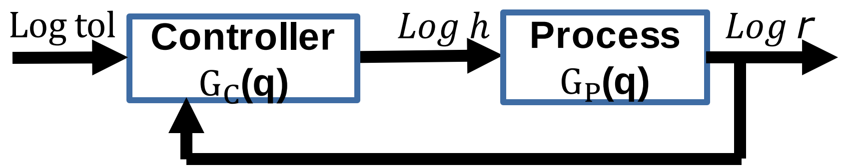
$log h_n = G_{c1}(q)(log\ tol - log r_n), \qquad G_{c1}(q) = k_I \frac{q}{q-1}$

$log r_n = G_{tol}(q) log\ tol + G_\phi(q) log\|\phi_n\|$

If h_n is small, $G_{tol}(q) = \frac{kk_I}{q-1+kk_I}, \qquad G_\phi(q) = \frac{q-1}{q(q-1+kk_I)}$	If h_n is increased, $G_{tol}(q) = \frac{k_I((C_1-1)q+1-C_1+C_2)}{q^2+(-2+k_I(C_1-1))q+1+k_I(1-C_1+C_2)}$ $G_\phi(q) = \frac{(q-1)((C_1-1)q+1-C_1+C_2)}{q^2+(-2+k_I(C_1-1))q+1+k_I(1-C_1+C_2)}$

시간 적응 기법 설계

- **Step Size Control for Runge-Kutta**
  - **The Closed Loop Using PI Controller**



- $h_n = (\frac{tol}{r_n})^{k_I}(\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$

$$\log h_n = G_{c2}(q)(\log tol - \log r_n), \qquad G_{c2}(q) = k_I \frac{q}{q-1} + k_P = \frac{(k_I + k_P)q - k_P}{q-1}$$

$$\log r_n = G_{tol}(q) \log tol + G_{\phi}(q) \log \|\phi_n\|$$

The denominators are given below

$$G_{p1}(q):\ q^2 + (-1 + k(k_I + k_P))q - kk_P = 0$$

$$G_{p2}(q):\ q^3 + (-2 + C_1(k_I + k_P))q^2 + (1 + C_2(k_I + k_P) - C_1(k_I + 2k_P))q + k_P(C_1 - C_2) = 0$$

$$G_{p3}(q):\ q^3 + (-2 + (C_1 - 1)(k_I + k_P))q^2 + (1 + C_2(k_I + k_P) + (1 - C_1)(k_I + 2k_P))q + k_P(C_1 - C_2 - 1) = 0$$

$$k_I = \frac{0.3}{k}, \quad k_P = \frac{0.4}{k}$$

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

■ Error analysis of system equations

$$Y' = \frac{dY}{dt} = F(t;Y), \qquad \textit{with the initial value } Y(a) = Y_0 \ (a \le Y \le b)$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, F(t;Y) = \begin{bmatrix} f_1(t;y_1,y_2,\dots,y_m) \\ f_2(t;y_1,y_2,\dots,y_m) \\ \vdots \\ f_m(t;y_1,y_2,\dots,y_m) \end{bmatrix}$$

$Y(t)$: *exact solution*
Y_n^*: *theoretical solution*
Y_n: *computed solution*

$$Y_{n+1}^* = Y_n^* + \frac{h}{6}\sum_{j=1}^{4} \gamma_j K_j^*$$

$$K_j^* = F(t_n + \delta_j h, Y_n^* + h\delta_j K_{j-1}^*)$$

$$Y_{n+1} = Y_n + \frac{h}{6}\sum_{j=1}^{4} \gamma_j K_j + R_n$$

$$K_j = F(t_n + \delta_j h, Y_n + h\delta_j K_{j-1})$$

We can get the error by subtracting *exact solution* from *computed solution.*

$$Y_{n+1} - Y(t_{n+1}) = Y_{n+1} - Y(t_n + h)$$

$$= Y_n - Y(t_n) + \frac{h}{6}\sum_{j=1}^{4} \gamma_j (K_j - K(j)) + R$$

$$e_{n+1} = e_n + \frac{h}{6}\sum_{j=1}^{4} \gamma_j (K_j - K(j)) + R$$

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

- Error analysis of system equations

$$e_{n+1} = e_n + \frac{h}{6}\sum_{j=1}^{4} \gamma_j (K_j - K(j)) + R$$

Assumption 1: $F(t;Y)$ is continuous function of Y in the interval
Assumption 2: Jacobian matrix $(\frac{\partial f_i}{\partial y_j})$ exists in the interval whose endpoints are Y_n and $Y(t_n)$

$$K_1 - K(1) = \left(\frac{\partial f_i}{\partial y_j}\right) e_n$$

$$K_2 - K(2) = \left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\right\} e_n$$

$$K_3 - K(3) = \left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\right\}\right\} e_n$$

$$K_4 - K(4) = \left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + h\left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\left\{E + \frac{h}{2}\left(\frac{\partial f_i}{\partial y_j}\right)\right\}\right\}\right\}\right\} e_n$$

시간 적응 기법 설계

Step Size Control for Runge-Kutta

Error analysis of system equations

$$e_{n+1} = \left[E + \frac{h}{6}\left\{\left(\frac{\partial f_i}{\partial y_j}\right) + 2\left(\frac{\partial f_i}{\partial y_j}\right) + 2\left(\frac{\partial f_i}{\partial y_j}\right) + \left(\frac{\partial f_i}{\partial y_j}\right)\right\} + \frac{h^2}{6}\left\{\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right) + \left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right) + \left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\right\} + \frac{h^3}{12}\left\{\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right) + \left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\right\} + \frac{h^4}{24}\left\{\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\left(\frac{\partial f_i}{\partial y_j}\right)\right\}\right] e_n + R$$

Assume that R and matrix $\left(\frac{\partial f_i}{\partial y_j}\right)$ are constant in a small interval (t_n, t_{n+1})

$$e_{n+1} = \left(\sum_{v=0}^{4} \frac{h^v}{v!} J^v\right) e_n + R$$

Transform to the classical canonical form -> $\hat{e} = T^{-1}e$

Ignore R to make the equation be homogeneous

$$\hat{e}_{n+1} = \left(\sum_{v=0}^{4} \frac{h^v}{v!} K^v\right) \hat{e}_n \quad with\ K = T^{-1}JT$$

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

- Error analysis of system equations

$$\hat{e}_{n+1} = \left(\sum_{v=0}^{4} \frac{h^v}{v!} K^v \right) \hat{e}_n \quad with\ K = T^{-1} J T$$

If matrix J is full rank,

Matrix K become diagonal

$$e_{n+1} = \begin{pmatrix} \sum \frac{(h\rho_1)^v}{v!} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum \frac{(h\rho_m)^v}{v!} \end{pmatrix} \mathrm{e_n}$$

$$\boxed{\hat{e}_n = diag\left(\left\{ \sum \frac{(h\rho_i)^v}{v!} \right\}^n \right) \times C}$$

If matrix J isn't full rank, matrix J cannot be diagonalized.

Some other way must be considered for transformation.

Stability region analysis is based on complex plane.
But, complex properties come from eigenvalues only.

$$matrix\ J \in M_{10}(\mathbb{R})$$

-> matrix K may be transformed to Jordan form

For a matrix on a general scalar field $\mathrm{F}(\mathbb{R})$, Jordan decomposition exists when minimal polynomial is completely decomposed into linear equations in $\mathrm{F}(\mathbb{R})$.

시간 적응 기법 설계

- **Step Size Control for Runge-Kutta**
  - Error analysis of system equations
    - Transformation to Jordan form
      - matrix K has submatrix in the leading diagonal

$$K = \begin{bmatrix} K_{11} & & & & 0 \\ & K_{12} & & & \\ & & \ddots & & \\ & & & K_{ij} & \\ 0 & & & & \ddots \end{bmatrix}, \quad with\ K_{ij} = \begin{bmatrix} \rho_i & 1 & & 0 \\ & \rho_i & \ddots & \\ & & \ddots & 1 \\ 0 & & & \rho_i \end{bmatrix}$$

$$The\ number\ of\ K_{ij} = The\ number\ of\ independent\ eigenvalues$$
$$K_{ij} \in M_l(\mathbb{C}), l = algebric\ multiplicity$$

$$\hat{e}_{n+1} = \left(\sum_{v=0}^{4} \frac{h^v}{v!} K_{\sigma\gamma}^v \right) \hat{e}_n, \quad with\ K_{\sigma\gamma}^v = \begin{bmatrix} \rho_\sigma^v & \binom{v}{1} \rho_\sigma^{v-1} & \cdots & \binom{v}{\gamma-1} \rho_\sigma^{v-\gamma+1} \\ 0 & \rho_\sigma^v & \cdots & \binom{v}{\gamma-2} \rho_\sigma^{v-\gamma+2} \\ & & \ddots & \\ 0 & & & \rho_\sigma^v \end{bmatrix}$$

시간 적응 기법 설계

- **Step Size Control for Runge-Kutta**
  - Error analysis of system equations
    - Transformation to Jordan form(contd.)

    - $$\hat{e}_{n+1} = \begin{bmatrix} \sum_{v=0}^{4}\frac{(h\rho_\sigma)^v}{v!} & \frac{h}{1!}\sum_{v=0}^{3}\frac{(h\rho_\sigma)^v}{v!} & \frac{h^2}{2!}\sum_{v=0}^{2}\frac{(h\rho_\sigma)^v}{v!} & & \frac{h^4}{4!} & \cdots & 0 \\ 0 & \sum_{v=0}^{4}\frac{(h\rho_\sigma)^v}{v!} & & \cdots & & & \\ 0 & 0 & \sum_{v=0}^{4}\frac{(h\rho_\sigma)^v}{v!} & & & & \\ & \vdots & & \ddots & & \vdots & \\ & & & \cdots & & \sum_{v=0}^{4}\frac{(h\rho_\sigma)^v}{v!} & \\ 0 & & & & & & \sum_{v=0}^{4}\frac{(h\rho_\sigma)^v}{v!} \end{bmatrix} \hat{e}_n$$

$$\boxed{\hat{e}_{i,(n+1)} = \sum_{j=0}^{\gamma-1}\frac{h^j}{j!}\sum_{v=0}^{4-j}\frac{(h\rho_\sigma)^v}{v!}\hat{e}_{(i+j),n} \quad with\ (i = 1, 2, \ldots, \gamma;\ 0 \le j \le 4)}$$

시간 적응 기법 설계

Step Size Control for Runge-Kutta

Error analysis of system equations

$$\hat{\boldsymbol{e}}_{\boldsymbol{n}+\mathbf{1}} = \left(\sum_{\boldsymbol{v}=\mathbf{0}}^{\mathbf{4}} \frac{\boldsymbol{h}^{\boldsymbol{v}}}{\boldsymbol{v}!} \boldsymbol{K}^{\boldsymbol{v}}\right) \hat{\boldsymbol{e}}_{\boldsymbol{n}} + \widehat{\boldsymbol{R}}$$

- **Homogeneous solution**
  - $\hat{e}_n = diag\left(\left\{\sum \frac{(h\rho_1)^v}{v!}\right\}^n\right) \times C$
  - $\hat{e}_{i,(n+1)} = \sum_{j=0}^{\gamma-1} \frac{h^j}{j!} \sum_{v=0}^{4-j} \frac{(h\rho_\sigma)^v}{v!} \widehat{e}_{(i+j),n} \quad with\ (i = 1,2, \dots, \gamma; 0 \le j \le 4)$
- **Particular solution**

$$\left(\boldsymbol{E} - \sum_{\boldsymbol{v}=\mathbf{0}}^{\mathbf{4}} \frac{\boldsymbol{h}^{\boldsymbol{v}}}{\boldsymbol{v}!} \boldsymbol{J}^{\boldsymbol{v}}\right) \boldsymbol{e}_{\boldsymbol{n}} = \boldsymbol{R}$$

$$\boldsymbol{e}_{\boldsymbol{n}} = \left(\boldsymbol{E} - \sum_{\boldsymbol{v}=\mathbf{0}}^{\mathbf{4}} \frac{\boldsymbol{h}^{\boldsymbol{v}}}{\boldsymbol{v}!} \boldsymbol{J}^{\boldsymbol{v}}\right)^{-1} \boldsymbol{R}$$

$$\therefore \boldsymbol{e}_{\boldsymbol{n}} = \mathrm{T} \begin{bmatrix} \mathrm{c}_{11} & & 0 \\ & \ddots & \\ 0 & & \mathrm{c}_{\mathrm{mm}} \end{bmatrix} \begin{bmatrix} \left\{\sum \frac{(h\rho_1)^v}{v!}\right\}^n \\ \vdots \\ \left\{\sum \frac{(h\rho_\mathrm{m})^v}{v!}\right\}^n \end{bmatrix} - \left(\sum_{\boldsymbol{v}=\mathbf{1}}^{\mathbf{4}} \frac{\boldsymbol{h}^{\boldsymbol{v}}}{\boldsymbol{v}!} \boldsymbol{J}^{\boldsymbol{v}}\right)^{-1} \boldsymbol{R}$$

시간 적응 기법 설계

Step Size Control for Runge-Kutta

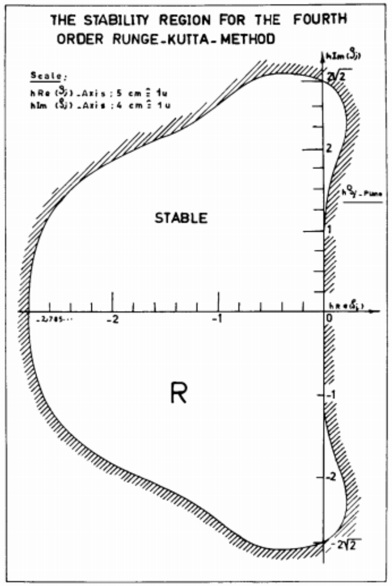
- **Error analysis of system equations**

$$\boldsymbol{e_n} = \mathrm{T}\begin{bmatrix} c_{11} & & 0 \\ & \ddots & \\ 0 & & c_{mm} \end{bmatrix} \begin{bmatrix} \left\{\sum \frac{\left(h_1\rho_{1,1}\right)^v}{v!}\right\} \cdots \left\{\sum \frac{\left(h_{n-1}\rho_{1,n-1}\right)^v}{v!}\right\} \\ \vdots \\ \left\{\sum \frac{\left(h_1\rho_{m,1}\right)^v}{v!}\right\} \cdots \left\{\sum \frac{\left(h_{n-1}\rho_{m,n-1}\right)^v}{v!}\right\} \end{bmatrix} - \left(\sum_{v=1}^{4} \frac{\boldsymbol{h}^{\boldsymbol{v}}}{\boldsymbol{v}!}\boldsymbol{J}^{\boldsymbol{v}}\right)^{-1} \boldsymbol{R}$$

  - **Runge-Kutta 4th order method for system equations is said to be stable when $\left\{\sum \frac{(h\rho_i)^v}{v!}\right\}$ is in the unit circle in the complex plane.**

$$\left|\sum \frac{(h\rho_i)^v}{v!}\right| < 1$$

  - **Stability region**
    - **Stability region of RK4 for system equations is similar to that for the 1D ODE.**



시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

■ Error analysis of system equations

$$e_n = T\begin{bmatrix} c_{11} & & 0 \\ & \ddots & \\ 0 & & c_{mm} \end{bmatrix}\begin{bmatrix} \left\{\sum \frac{\left(h_1\rho_{1,1}\right)^v}{v!}\right\} \cdots \left\{\sum \frac{\left(h_{n-1}\rho_{1,n-1}\right)^v}{v!}\right\} \\ \vdots \\ \left\{\sum \frac{\left(h_1\rho_{m,1}\right)^v}{v!}\right\} \cdots \left\{\sum \frac{\left(h_{n-1}\rho_{m,n-1}\right)^v}{v!}\right\} \end{bmatrix} - \left(\sum_{v=1}^{4} \frac{h^v}{v!} J^v\right)^{-1} R$$

$$i\ corresponding\ to\ the\ maximum\ error(e_{max})$$

1. It is mathematically reasonable to control the stepsize with r_n which maximizes $\left|\sum \frac{(h\rho)^v}{v!}\right|$, rather than based on e_{max}
2. Process transfer function for 1D ODE is identical for system equations
   => Controller coefficient k_I, k_P for 1D ODE can be used.

시간 적응 기법 설계

◆ Step Size Control for Runge-Kutta

1.
Application to 1D differential equations

$$log\ r_n = \frac{G_p G_c}{1 + G_p G_c} log\ h_n$$

$$k_I = \frac{0.3}{k},\ k_P = \frac{0.4}{k}$$

(k is order or error)

2.
Error analysis of system differential equations

$$e_n = T \begin{bmatrix} c_{11} & & 0 \\ & \ddots & \\ 0 & & c_{mm} \end{bmatrix} \begin{bmatrix} \left\{ \sum \frac{(h_1 \rho_{1,1})^v}{v!} \right\} \cdots \left\{ \sum \frac{(h_{n-1} \rho_{1,n-1})^v}{v!} \right\} \\ \vdots \\ \left\{ \sum \frac{(h_1 \rho_{m,1})^v}{v!} \right\} \cdots \left\{ \sum \frac{(h_{n-1} \rho_{m,n-1})^v}{v!} \right\} \end{bmatrix}$$

The stability region of step size

$$\left| \sum \frac{(h_j \rho_{i,j})^v}{v!} \right| < 1$$

3.
Application to system differential equations

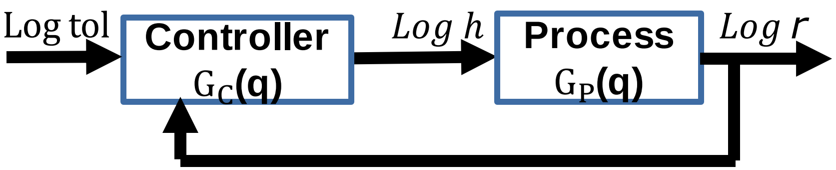
$$h_n = (\frac{tol}{r_n})^{k_I} (\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$$

(1) Control coefficient for 1D differential equation is still used.
(2) Error r_n is calculated by Richardson's extrapolation
(3) PI controller is switched to integral controller when any eigenvalue of system equations us outside the stability region.

시간 적응 기법 설계

Determining Controller for Stepsize Control

The Closed Loop Using PI Controller



- $h_n = (\frac{tol}{r_n})^{k_I} (\frac{r_{n-1}}{r_n})^{k_P} h_{n-1}$

LU-SGS : $k_I = 0.175,\ k_P = 0.075$

Runge-Kutta method : $k_I = \frac{0.3}{k}$, $k_P = \frac{0.4}{k}$ $(k = O(leading\ error))$

- **Prevent unnecessary oscillation**

$$\frac{h_n}{h_{n-1}} = \begin{cases} 2 & (\frac{h_n}{h_{n-1}} \geq 2) \\ \frac{h_n}{h_{n-1}} & ELSE \\ 1 & (1 \leq \frac{h_n}{h_{n-1}} \leq 1.2) \end{cases}$$

시간 적응 기법 적용

Application to 6DOF Equation

Jacobian of 6DOF Equation

$$
J = \begin{bmatrix}
J_{11} & J_{12} & J_{13} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
J_{21} & J_{22} & J_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
J_{31} & J_{32} & J_{33} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -w & v & 0 & r & -q & 0 & 0 & 0 & 0 \\
w & 0 & -u & -r & 0 & p & 0 & 0 & 0 & 0 \\
-v & u & 0 & q & -p & 0 & 0 & 0 & 0 & 0 \\
-\frac{q_1}{2} & -\frac{q_2}{2} & -\frac{q_3}{2} & 0 & 0 & 0 & 0 & -\frac{\mathrm{p}}{2} & -\frac{\mathrm{q}}{2} & -\frac{\mathrm{r}}{2} \\
\frac{q_0}{2} & -\frac{q_3}{2} & \frac{q_2}{2} & 0 & 0 & 0 & \frac{\mathrm{p}}{2} & 0 & \frac{\mathrm{r}}{2} & -\frac{\mathrm{q}}{2} \\
\frac{q_3}{2} & \frac{q_0}{2} & -\frac{q_1}{2} & 0 & 0 & 0 & \frac{\mathrm{q}}{2} & -\frac{\mathrm{r}}{2} & 0 & \frac{\mathrm{p}}{2} \\
-\frac{q_2}{2} & \frac{q_1}{2} & \frac{q_0}{2} & 0 & 0 & 0 & \frac{\mathrm{r}}{2} & \frac{\mathrm{q}}{2} & -\frac{\mathrm{p}}{2} & 0
\end{bmatrix}
$$

$$J_{k1} = \{a_{k1}(rI_{21} - qI_{31}) + a_{k2}(2pI_{31} - rI_{11}) + a_{k3}(qI_{11} - 2pI_{21})\} + \{a_{k2}I_{32}q - a_{k3}I_{22}q\} + \{a_{k2}I_{33}r - a_{k3}I_{23}r\}$$

$$J_{k2} = \{-a_{k1}I_{31}p + a_{k3}I_{11}p\} + \{a_{k1}(rI_{22} - 2qI_{32}) + a_{k2}(pI_{32} - rI_{12}) + a_{k3}(2qI_{12} - pI_{22})\} + \{-a_{k1}I_{33}r + a_{k3}I_{13}r\}$$

$$J_{k3} = \{a_{k1}I_{21}p - a_{k2}I_{11}p\} + \{a_{k1}I_{22}q - a_{k2}I_{12}q\} + \{a_{k1}(2rI_{23} - qI_{33}) + a_{k2}(pI_{33} - 2rI_{13}) + a_{k3}(qI_{13} - pI_{23})\}$$

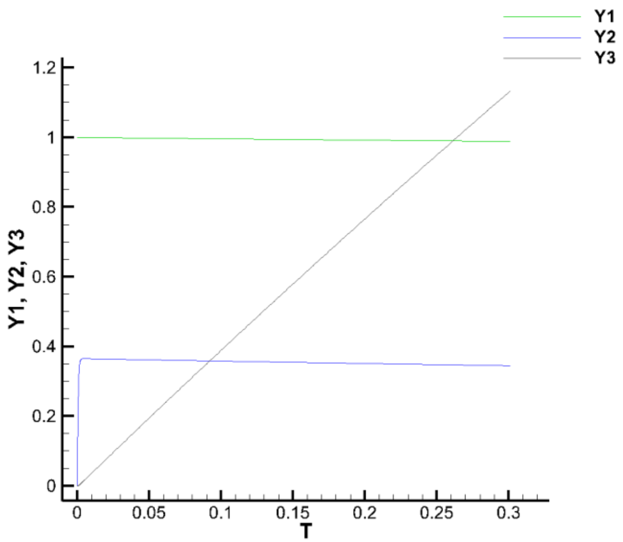
$$(K = 1,2,3)$$

Result. 3 equations

◆ **Time adaptation in system equation**

$$\dot{y}_1 = -0.04y_1 + 0.01y_2y_3 \qquad y_1(0) = 1$$
$$\dot{y}_2 = 400y_1 - 100y_2y_3 - 3000y_2^2 \qquad y_2(0) = 0$$
$$\dot{y}_3 = 30y_2^2 \qquad y_3(0) = 0$$

$$J = \begin{bmatrix} -0.04 & 0.01y_3 & 0.01y_2 \\ 400 & -6000y_2 - 100y_3 & -100y_2 \\ 0 & 60y_2 & 0 \end{bmatrix}$$

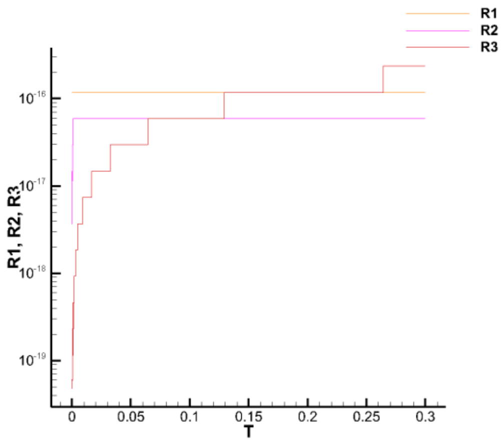


y_1 : Almost 1 in most areas
y_2 : Very transient in the beginning
y_3 : Increasing almost linearly

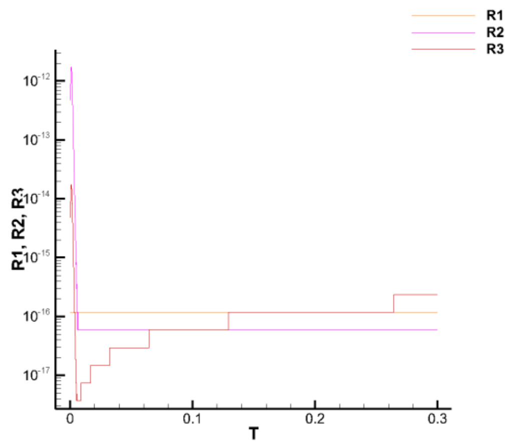
Result. 3 equations

◆ Time adaptation in system equation

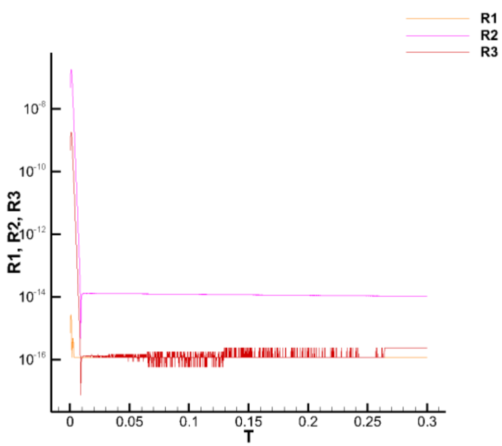
0.000001	0.00001	0.0001	0.001	0.01
5.92E-17	1.73E-12	1.62E-07	2.15E-2	NAN



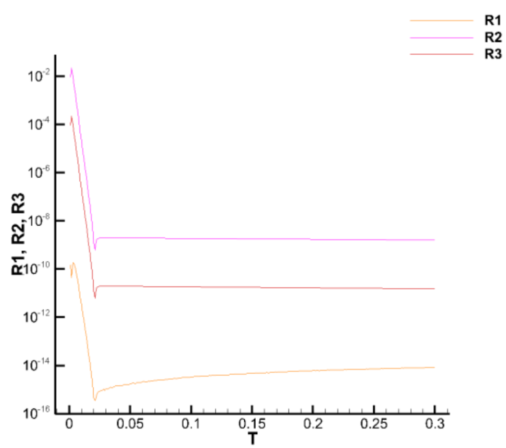
Reference($dt = 0.000001$)



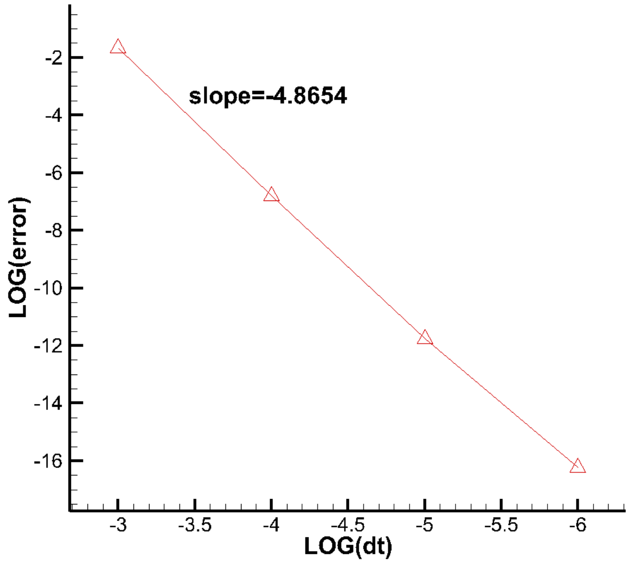
Reference($dt = 0.00001$)



Reference($dt = 0.0001$)



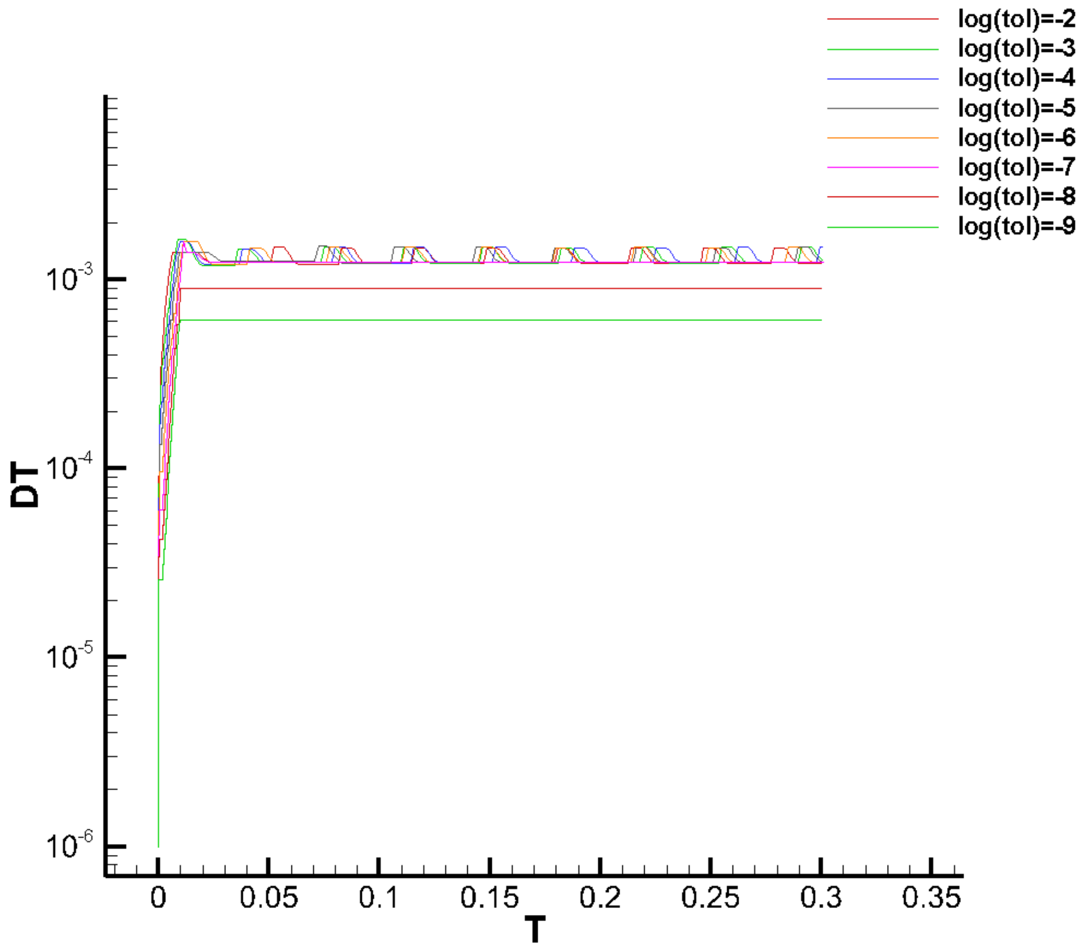
Reference($dt = 0.001$)



Result. 3 equations

- **Time adaptation in system equation**

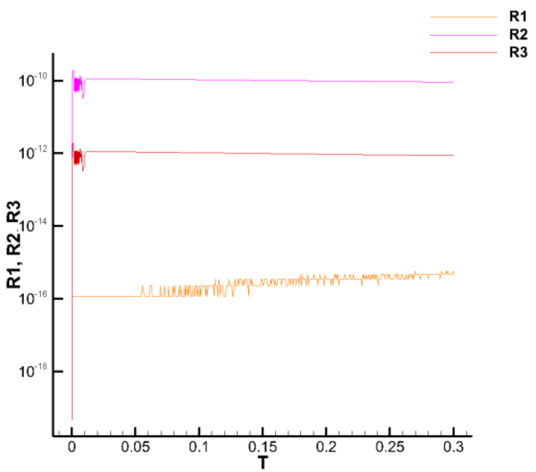
  $Starting\ DT = 0.000001$



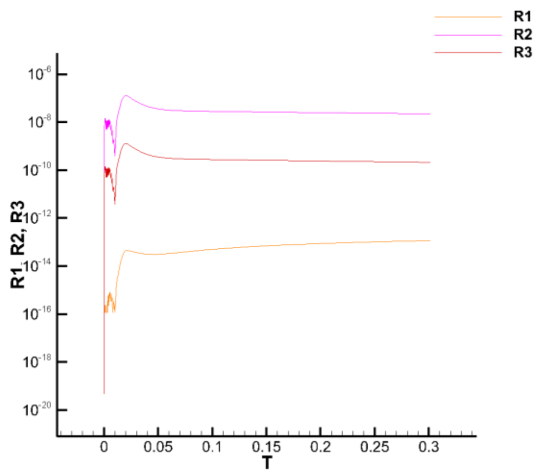
Result. 3 equations

- **Time adaptation in system equation**

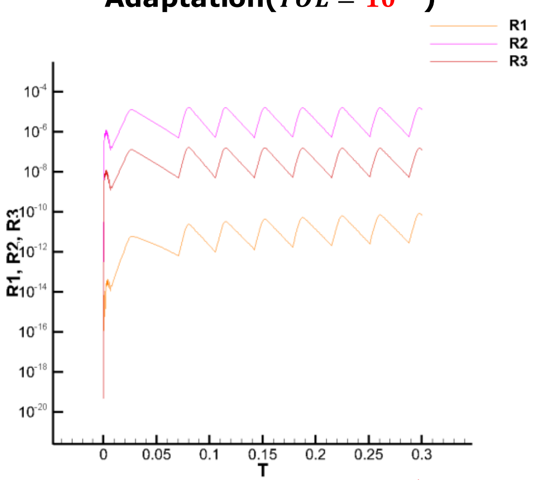
  $Starting\ DT = 0.000001$



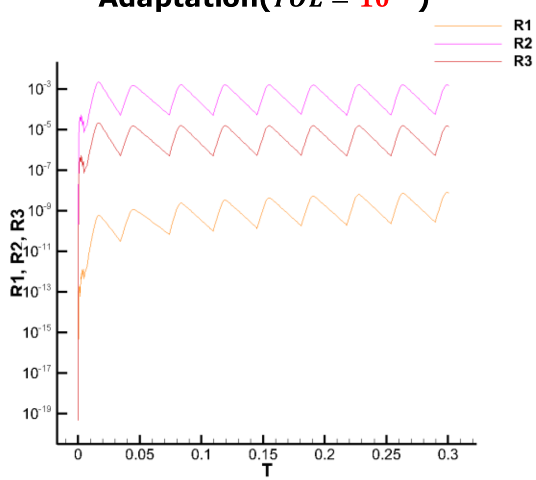
Adaptation($TOL = 10^{-9}$)



Adaptation($TOL = 10^{-7}$)



Adaptation($TOL = 10^{-5}$)



Adaptation($TOL = 10^{-3}$)

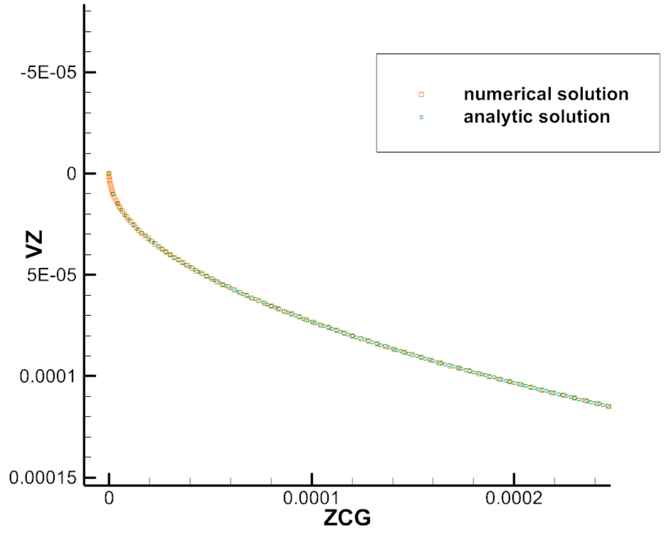
Reference : 300002 step
$TOL\ 10^{-9}$: 640 step
$TOL\ 10^{-7}$: 309 step
$TOL\ 10^{-5}$: 266 step
$TOL\ 10^{-3}$: 254 step

Result. 6DOF eq validation

Free-falling sphere

- **Comparison of numerical free-falling sphere velocity with analytic value, assuming that the air resistance is proportional to the square of the velocity**
  - $F(t) = -Mg + \frac{1}{2}\rho\pi r^2 C_D V(t)^2$
  - $V_{term} = \sqrt{(2Mg)(\rho\pi r^2 C_D)}$
  - $V(t) = \sqrt{V^2{}_{term}(1 - e^{\frac{2gz}{V^2{}_{term}}})}$

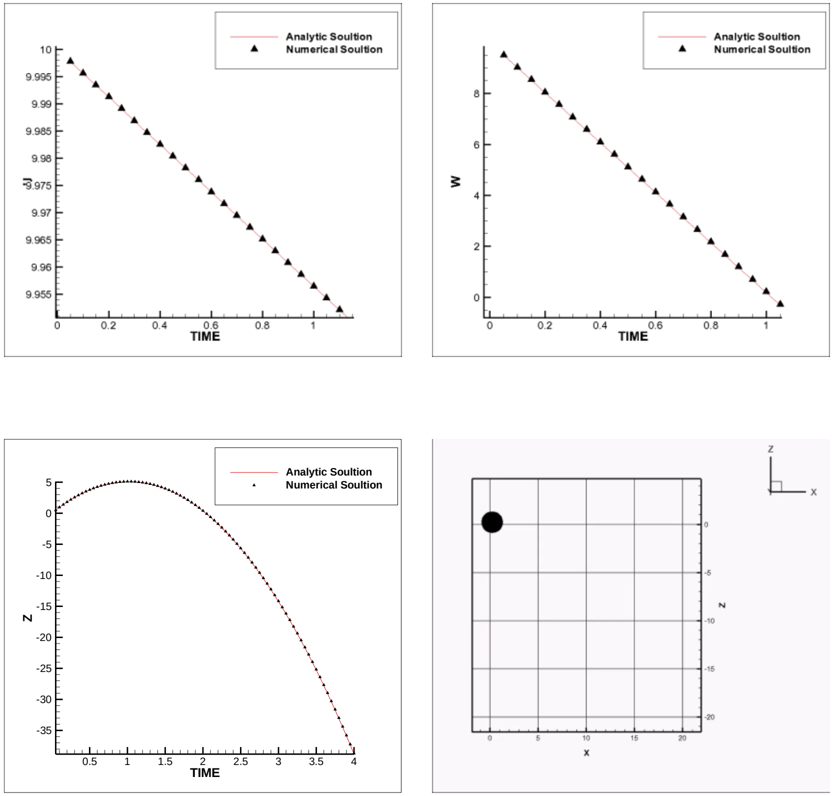
Analysis condition	
Drag	$D = 0.5\rho C_D A v^2, C_D = 0.5$
r	$1m$
g	$2.67 \times 10^{-5} m/s^2$
ρ	$1kg/m^3$



Result. 6DOF eq validation

- **Parabolic trajectory of sphere**
  - **Comparison of numerical parabolic motion trajectory with analytic value, assuming that the air resistance is proportional to the square of the velocity**

Analysis condition	
Drag	$D = 0.5\rho C_D A v^2, C_D = constant$
v_{x0}	10m/s
v_{y0}	0m/s
v_{z0}	10m/s



Result. 6DOF eq application

- **Spinning and tumbling cylinder**
  - No aerodynamic force, no gravity
  - Rotation axis is aligned with principal axis

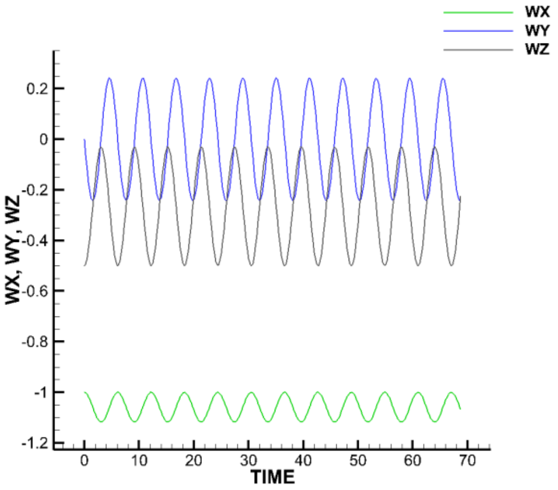
$$I_{(b)} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

$$I_{xx} = I_{yy}, \qquad I_{xx} - I_{zz} = \alpha I_{xx}$$

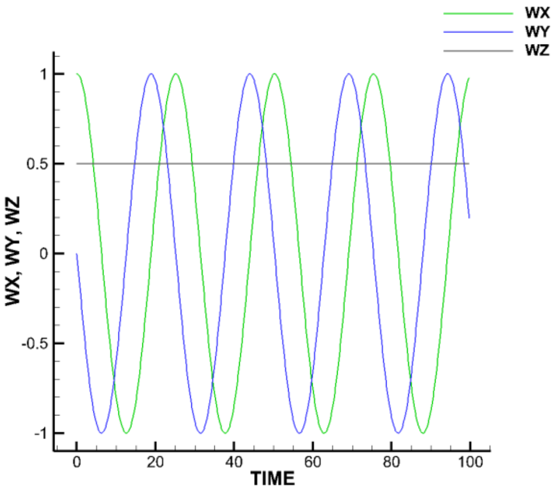
  - Exact solution exists

$$\omega_{x,(b)} = \mathrm{a}\cos(\lambda t)\,, \omega_{y,(b)} = \mathrm{b}\sin(\lambda t)\,, \omega_{z,(b)} = c$$

Analysis condition	
(I_{xx}, I_{yy}, I_{zz})	$(1, 1, 0.5)$
ω_{x0}	$1 rad/sec$
ω_{y0}	$0 rad/sec$
ω_{z0}	$0.5 rad/sec$



Angular velocity($Local\ frame$)

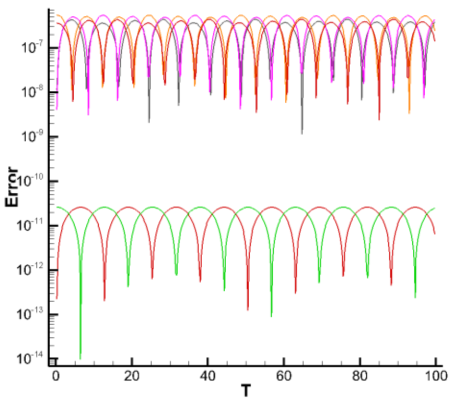


Angular velocity($Body\ frame$)

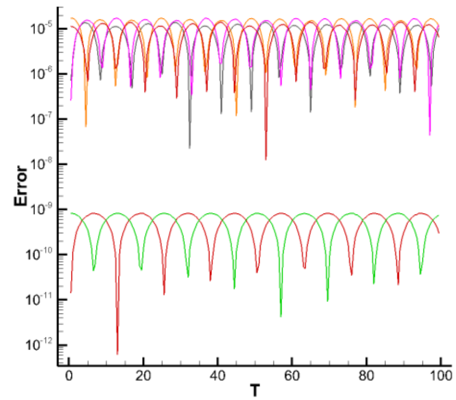
Result. 6DOF eq application

◆ Spinning and tumbling cylinder

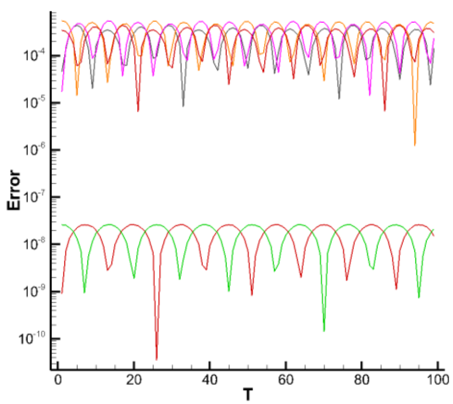
0.25	0.5	1.0	2.0	4.0
5.36E-07	1.68E-05	5.19E-04	1.54E-02	2.57E-01



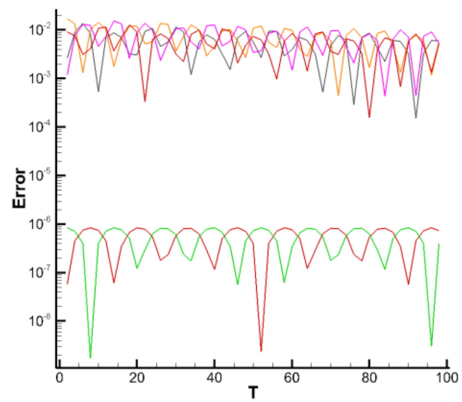
Reference($dt = 0.25$)



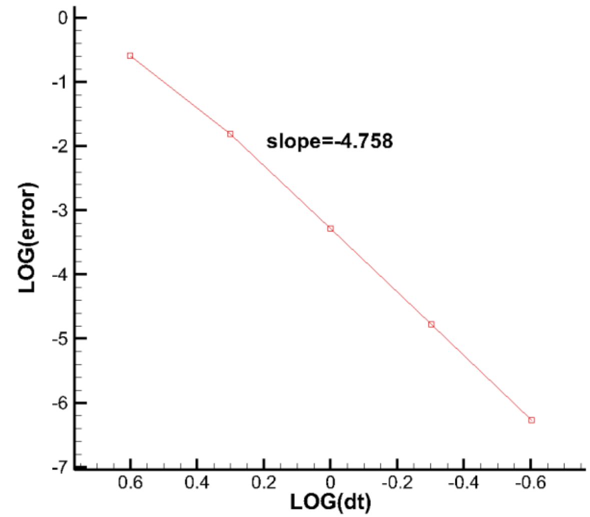
Reference($dt = 0.5$)



Reference($dt = 1.0$)

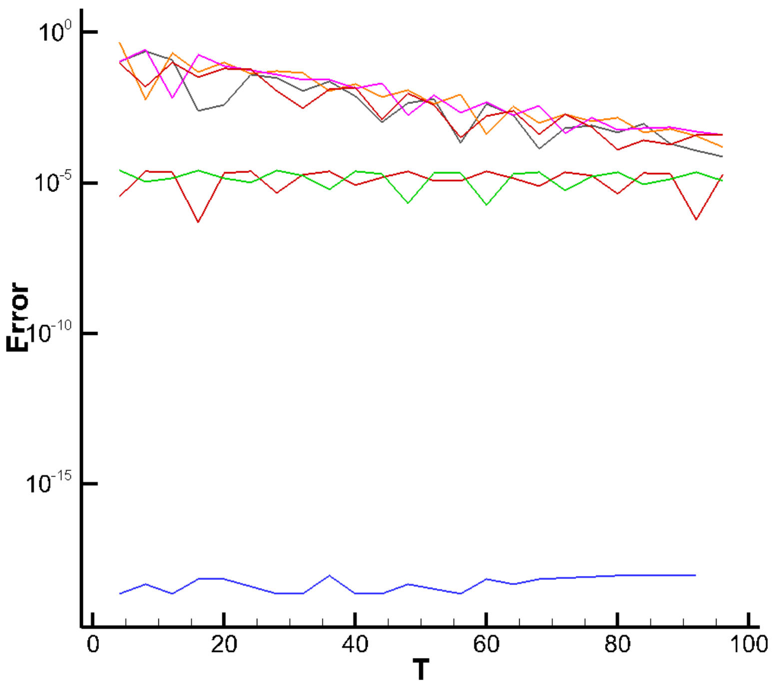


Reference($dt = 2.0$)

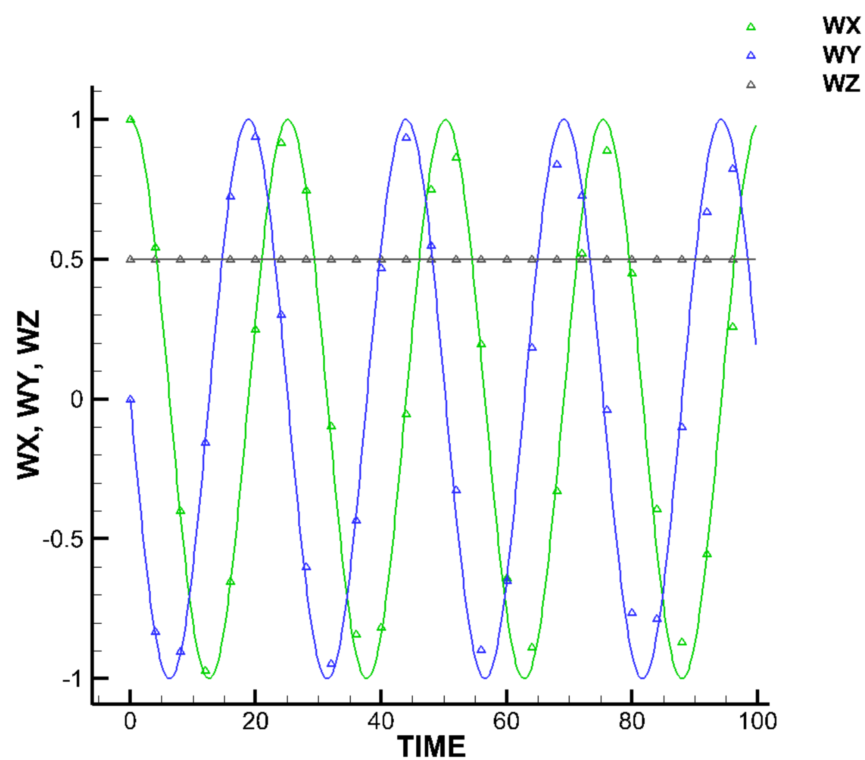


Result. 6DOF eq application

◆ **Spinning and tumbling cylinder**



Reference($dt = 4.0$ **)**

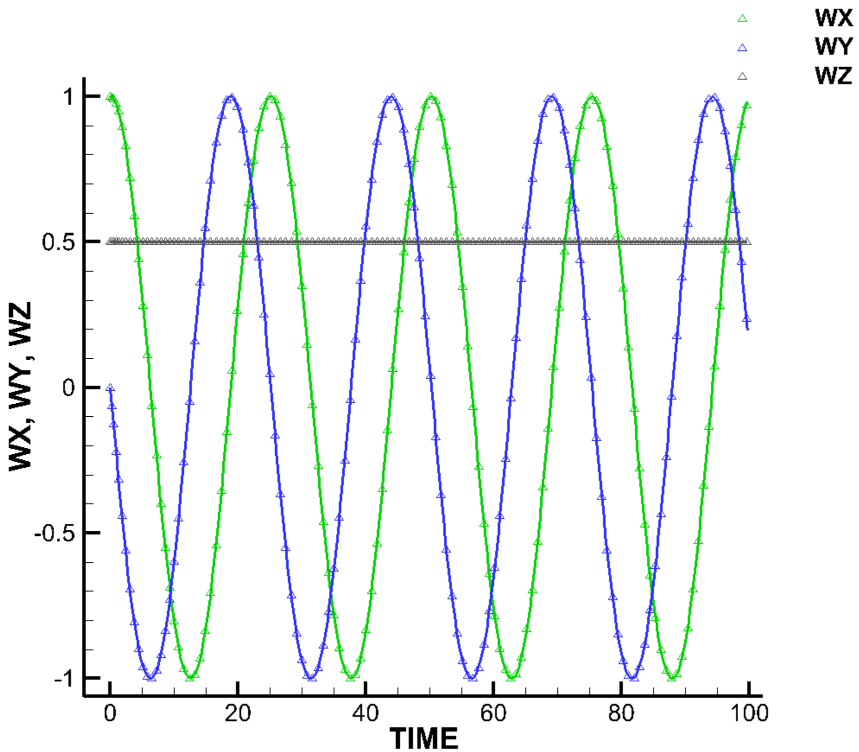


Deviation of trajectory($dt = 4.0$ **)**

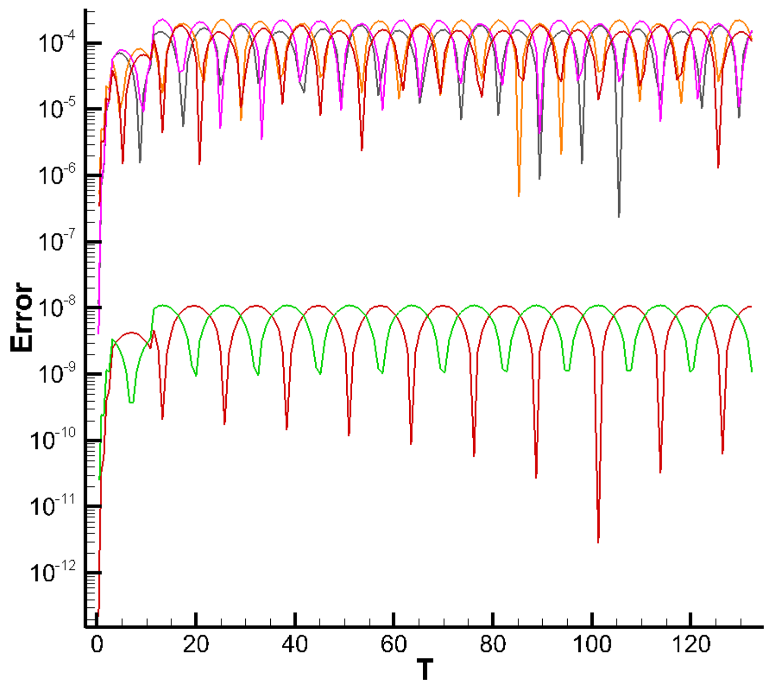
Result. 6DOF eq application

◆ Spinning and tumbling cylinder

$Starting\ DT = 0.25\ \&\ tolerance = 10^{-3}$



Trajectory in Adaptation($TOL = 10^{-3}$)



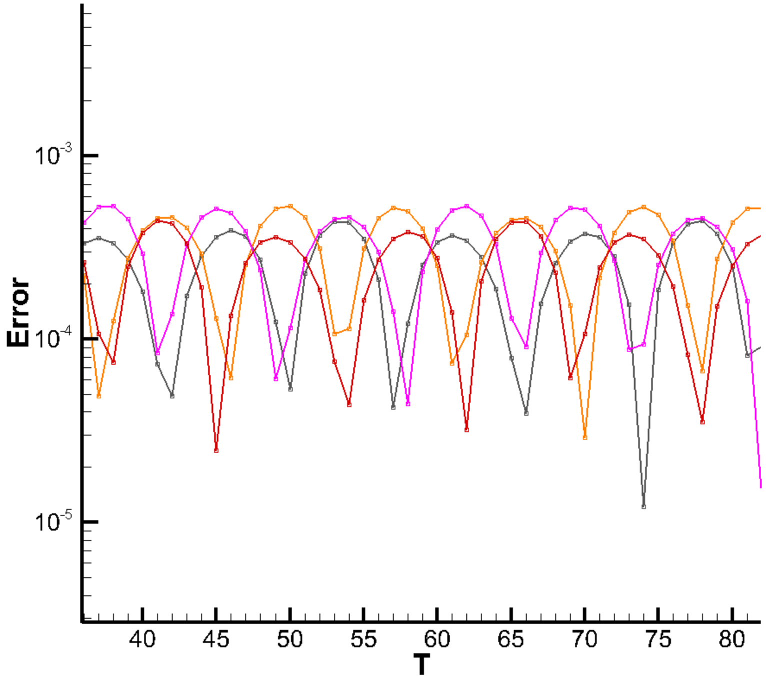
Error in Adaptation($TOL = 10^{-3}$)

Result. 6DOF eq application

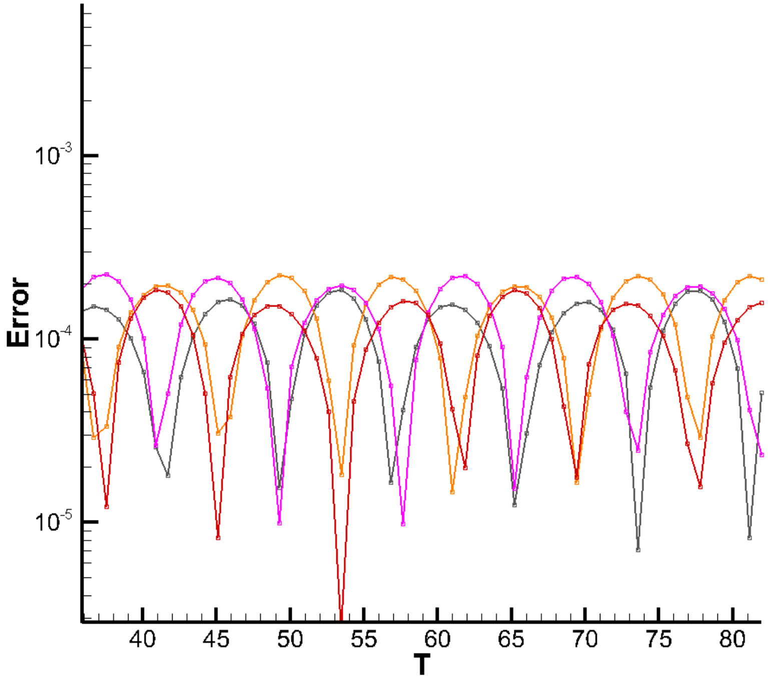
- **Spinning and tumbling cylinder**

$Starting\ DT = 0.25\ \&\ tolerance = 10^{-3}$

Reference : 400 step
$TOL\ 10^{-3}$: 164 step



Error in refence($\mathrm{dt} = 1.0$)

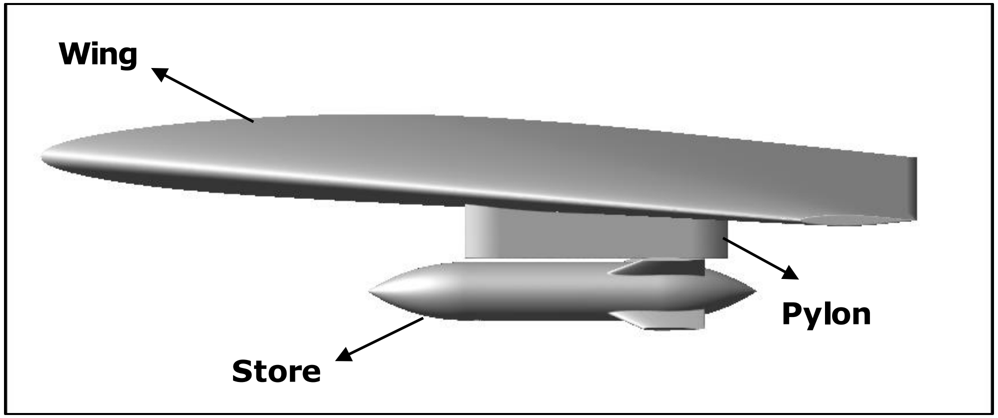


Error in Adaptation($TOL = 10^{-3}$)

Result. 6DOF eq application

◆ Eglin Store Separation Case

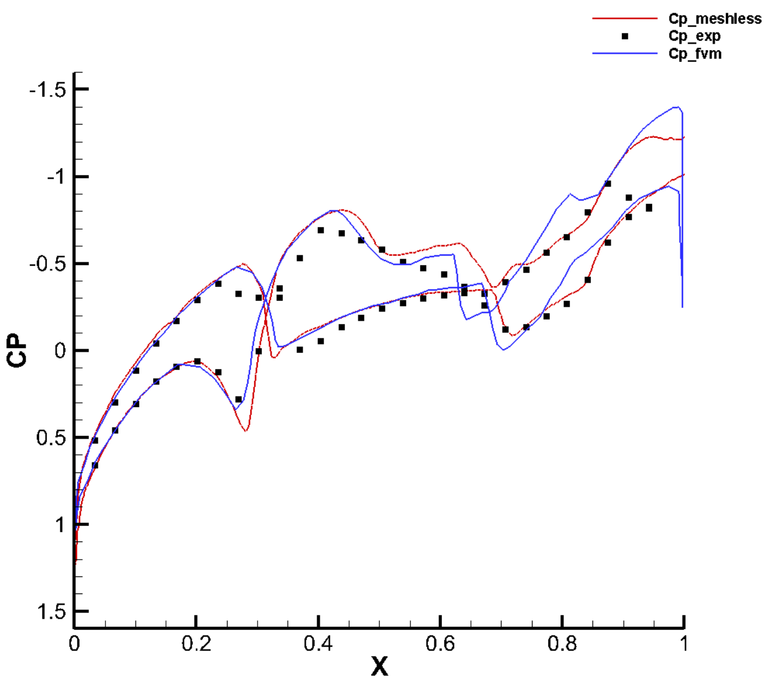
- EGLIN experiment was first conducted at Arnold Engineering Development Center in 1991.
- EGLIN is the standard store separation problem for validation of moving object trajectory.
  - Governing equation : ALE equation, 6 degree of freedom equation
  - Space discretization : AUSMPW+ with minmod TVD limiter
  - Time integration : LU-SGS for ALE eq, explicit Runge-Kutta method for 6dof eq.
  - Uniform time spacing $\Delta t = 0.001$



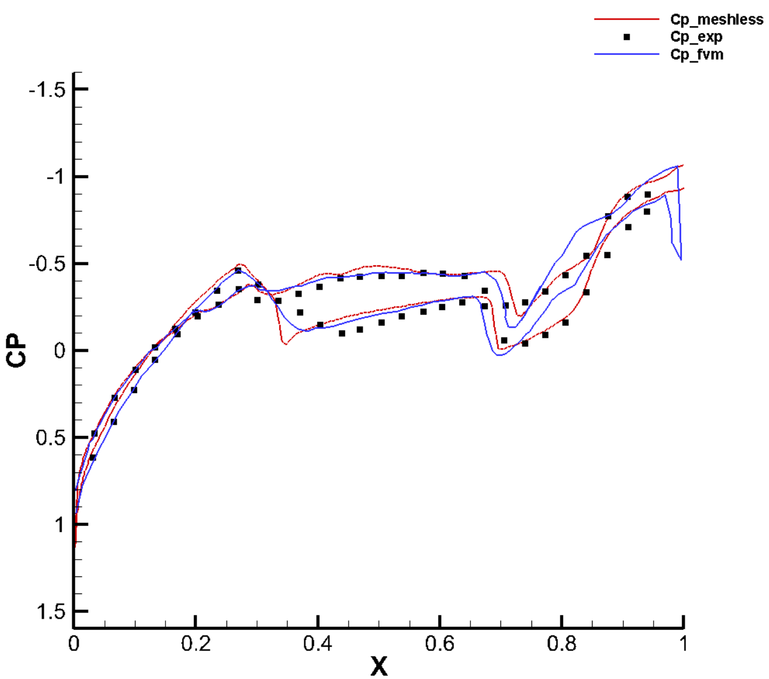
Flow condition
$M_\infty = 0.95$
$P_\infty = 35988.8Pa$
$T_\infty = 236.639K$

Result. 6DOF eq application

- **Eglin Store Separation Case**
  - **Coefficient of Pressure**
    - Coefficient of pressure (C_p) on the surface of store for the cross plane at angle 5 degree is shown.
    - C_p distribution of the store is compared with experiment and FVM. FVM values are the results presented by ANSYS at AIAA 2018 Scitech.



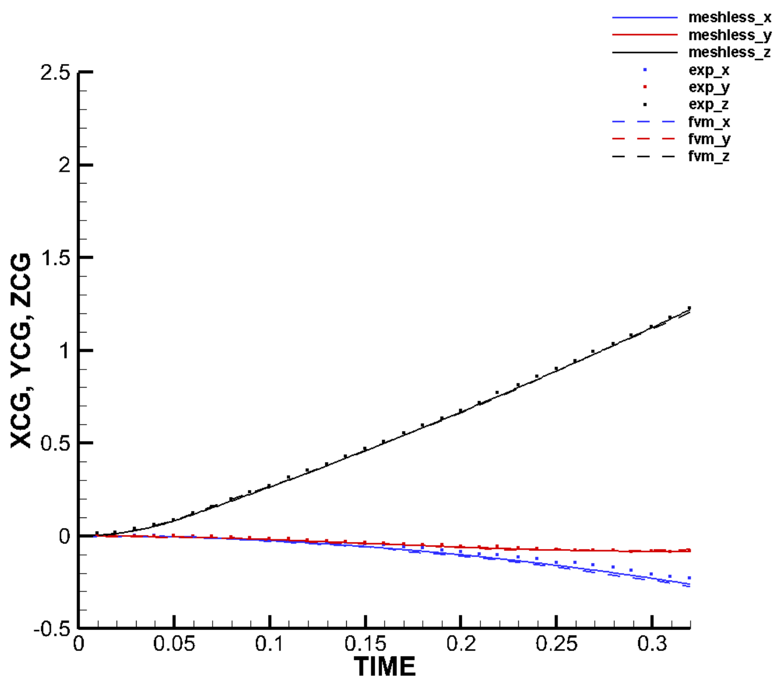
Cp distribution of the store (t=0sec)



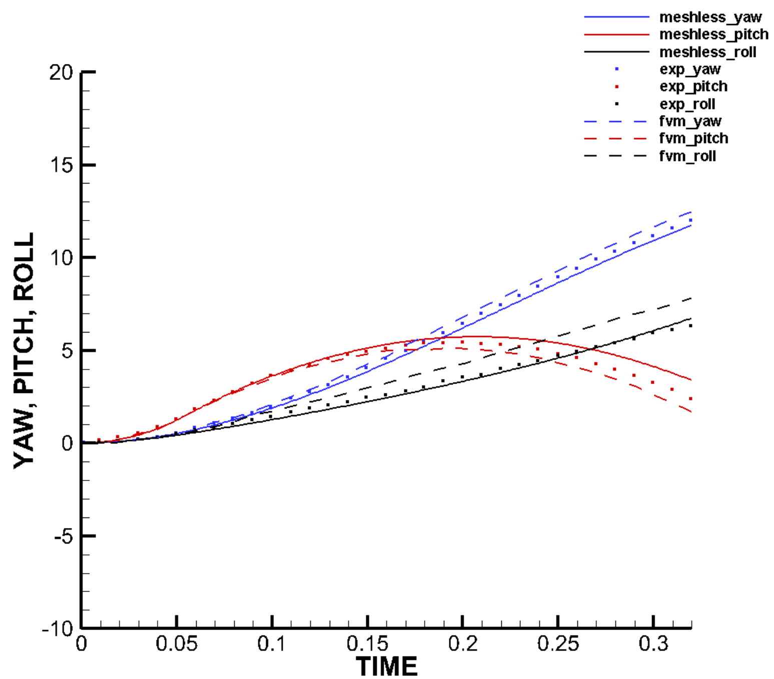
Cp distribution of the store (t=0.17sec)

Result. 6DOF eq application

- **Eglin Store Separation Case**
  - **Store Trajectory**
    - The trajectory of the store is also compared to the FVM and experiment results.
    - In EGLIN store separation case, it can be said that there are 3 types of forces - ejection force for safe store separation, gravity, aerodynamic force.
    - The numerical solution fits to experimental values well.



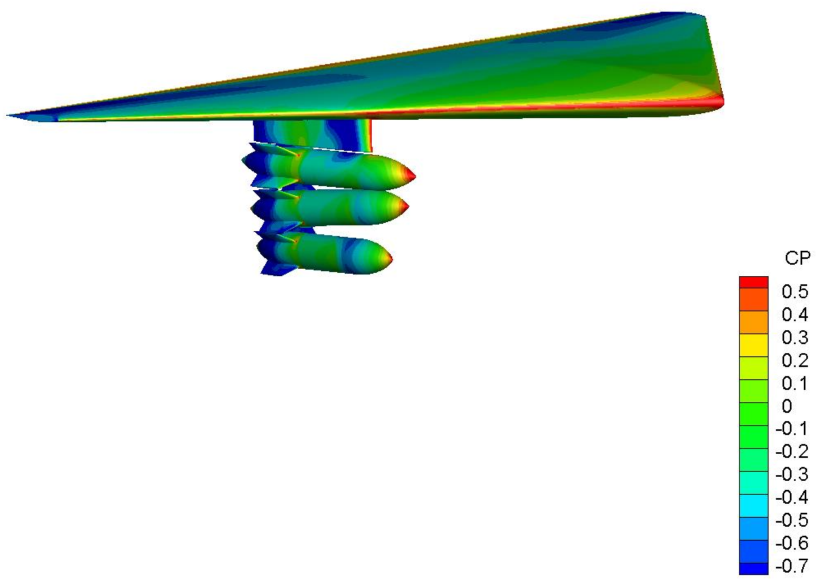
Center of Gravity Location



Angular Orientation

Result. 6DOF eq application

- **Eglin Store Separation Case**
  - **Store Trajectory**
    - **Figure below shows Location and posture of the store for 3 different time instances – 0sec, 0.17sec and 0.32sec.**



목 차

3. 결론

결론

Summary

Time adaptation algorithm using PI controller

- Application to LU-SGS & Dual time stepping
  - Instead of empirical formula, the newly defined control coefficient is used.
  - For robustness of inner iteration calculation, CFL number is controlled in a manner that solution change does not exceed the maximum allowable change.
  - Algorithm verification & validation
    - Sphere
    - ONERA M6 wing

- Runge-Kutta 4^{th} order method
  - Control coefficient for one-dimensional differential equation is chosen by Gustafsson.
  - The error of system differential equations is analyzed, and the stability region is obtained.
  - Application to system equation
  - Algorithm verification & validation
    - 3 equation problem
    - Free falling sphere & parabolic trajectory
    - Spinning & tumbling cylinder problem
    - EGLIN store separation

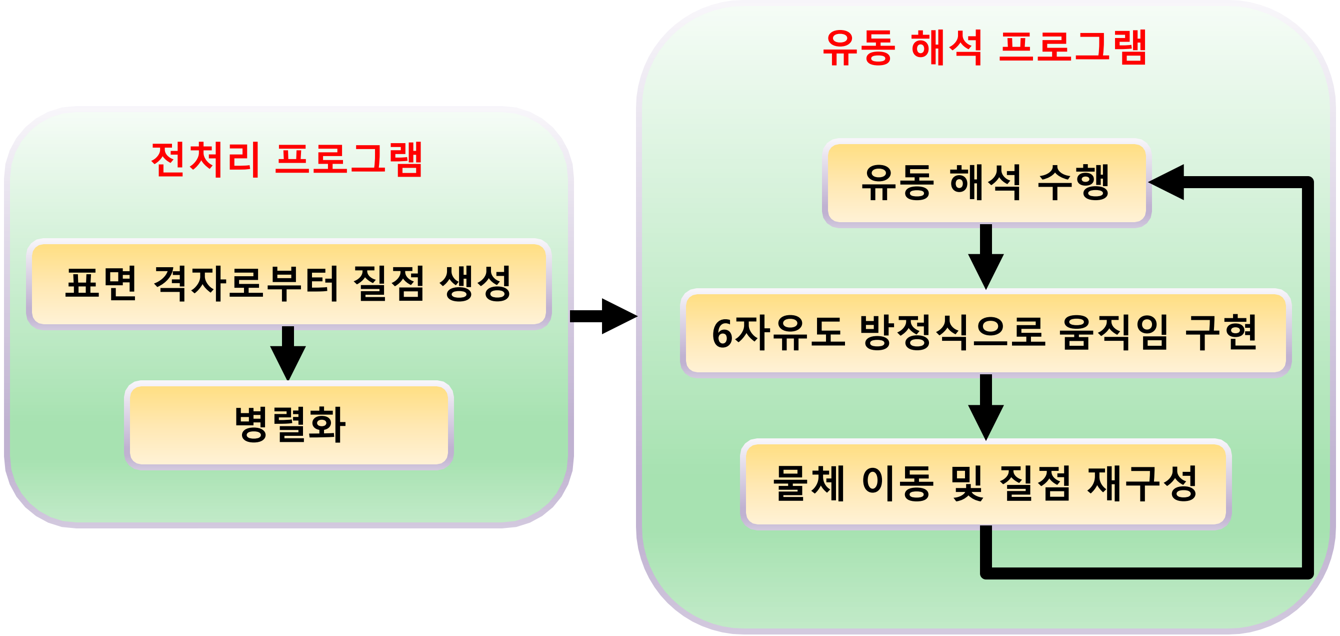
감사합니다

목 차

5. Appendix

공간 질점계 적응 기법

- 배경 질점계 구조 변경



공간 질점계 적응 기법

◆ **배경 질점계 구조 변경**