

## 적분행렬을 이용한 비선형 운동방정식 수치적분

이동헌\*, 권재욱\*\*, 최수진\*\*\*, 류동영\*\*\*\*, 주광혁\*\*\*\*\*

# Numerical Integration of Non-linear Equation of Motion using Operation of Integration

Donghun Lee\*, Jae-wook Kwon\*\*, Sujin Choi\*\*\*, Dong-young Rew\*\*\*\*, Gwanghyeok Ju\*\*\*\*\*

### Abstract

In this paper, numerical integration method using operational matrix of integration is studied. Using the operational matrix of integration, modified fixed point iteration method is introduced in order to solve rapidly an initial value problem for non-linear equation of motion. As an example, an initial value problem for orbital motion is considered. Through the numerical example, it is shown that the algorithm is efficient from the computational time point of view.

### 초 록

본 논문에서는 적분 행렬을 이용한 수치적분 방법을 연구하였다. 비선형 운동방정식에 대한 빠른 수치적분을 수행하기 위해, 적분 행렬을 이용한 개선된 fixed point iteration 방법을 소개한다. 예제로는 궤도 운동에 대한 수치적분 예를 고려한다. 수치 예제를 통하여, 본 논문에서 연구되는 알고리즘이 적분의 정밀도는 크게 저하시키지 않음과 동시에, 계산 시간 측면에서 효과적이라는 것을 보인다.

키워드 : 궤도 운동 (Orbital Motion), 수치적분 (Numerical integration), 르장드르 다항식 (Legendre Polynomial), 쿼드레처 룰(Quadrature Rule)

## 1. 서 론

수치적분은 비선형 운동방정식의 해석에 기본이 되는 방법으로 다양한 분야의 시뮬레이션과 해석에 널리 이용되고 있다. 수치적분 방법은 순차적으로 동일한 간격을 적분하는 방법과 적분오차를 고려하여 적분 간격을 조절하면서 적분을

수행하는 방법으로 나눌 수 있다. 또한 적분 방식에서 현재 시점과 동시에 미래 시점의 상태 변수의 고려 유무에 따라 implicit method와 explicit method 등 적분 방식에 따라 다양한 카테고리 나눌 수 있다. 한편, 운동방정식의 정밀한 적분을 위하여 고차식 수치적분 방법을 이용하기도 한다[1].

접수일(2014년 6월 16일), 수정일(1차 : 10월 24일), 게재 확정일(2014년 11월 1일)

\* 달탐사기술팀/donghlee@kari.re.kr

\*\* 달탐사체계팀/kjw@kari.re.kr

\*\*\* 달탐사체계팀/jin5864@kari.re.kr

\*\*\*\* 달탐사기술팀/dyrew@kari.re.kr

\*\*\*\*\* 달탐사체계팀/dyrew@kari.re.kr

수치적분에 있어 수치적분기의 중요한 성능지수는 정밀도와 계산 시간에 있다. 문제에 따라 수치적분의 정밀도를 증가시키기 위해서는 고차의 수치적분법을 사용하거나 적분 간격을 세밀하게 줄이게 되면, 계산량의 증가로 인한 부담이 계산 시간 증가로 이어져 수치해석의 결과를 도출하는데 오래 시간이 소요된다는 단점을 가지고 있다. 반면, 계산시간을 감소시키기 위해 저차식의 수치 적분법을 이용하거나, 적분간격을 증가시키는 경우에는 정밀도 저하가 우려되기도 한다 [1].

수치적분의 정밀도를 유지하면서, 계산 속도를 빠르게 하면, 다양한 효과를 볼 수 있을 것이다. 하나의 예로, 앞서 언급하였듯이, 수치적분은 다양한 분야에 적용되고 있는데, 비선형 상미분 운동방정식에 대한 해가 필요로 하는 궤적최적화 분야에도 수치적분은 널리 이용되고 있다. 특히, 궤적최적화 분야 중 제어입력을 이산화(parameterization of control input variables)하여, Runge-Kutta 등의 방식으로 수치적분을 수행한 뒤, 목적함수(Objective function)와 제한조건(Constraint)을 계산해야하는 direct shooting 기법에도 수치적분은 계산적 부담으로 작용한다.

최근, Picard iteration 또는 Fixed step iteration 기법과 같은 Quadrature rule을 이용한 수치적분 기법이 많이 연구되고 있다. 위 방법은 정밀도를 유지하면서, 적분시간을 효과적으로 줄이기 위해 많이 연구되고 있다. 참고 문헌에서는 Orthogonal Chebyshev 다항식과 Quadrature rule을 이용하여, 2-body 또는 Circular restricted three body dynamics에 대한 수치적분을 수행하는 연구를 수행하였다. 위 참고문헌에서는 기존의 Runge-Kutta 방법과 비교하였을 경우 제안된 방법이 정밀도를 유지하면서 계산 시간을 효과적으로 줄일 수 있다는 점을 나타내고 있다[2].

본 논문에서는 참고문헌 [2]의 연구 내용을 확장하여 연구를 수행하였다. 참고 문헌에서는 Orthogonal Chebyshev polynomial을 사용하였으나, 이와 달리 Orthogonal Legendre polynomial을 이용하였으며, 계산 시간의 효율성을 증대시키기 위해 적분행렬을 이용하였다.

Chebyshev polynomial과 Legendre polynomial의 차이는 적분 과정에서 사용되는 지점(Node)와 이에 따른 적분 가중치가 다르다. 그리고, 운동방정식에는 DE405 또는 DE421과 같은 천체력을 이용한 third body perturbation force 등을 고려하였으며, 위와 같은 조건에서 Fixed step iteration 기법을 적용하기 위한 방안을 연구하였다. 또한 단순한 Fixed step iteration 기법에서는 다항식의 차수가 고정되어 있는데, 이를 변화시키면서, 적분의 정밀도를 향상시키고자 하였다.

본 논문의 2장에서는 initial value problem (IVP)을 설명하고, 위 문제의 해를 구하기 위한 fixed point iteration (FPI)를 설명한다. 3장에서는 수치예제를 위한 궤도운동 방정식과 FPI의 개선된 사항을 수치 예제와 함께 소개한다.

## 2. 본 론

### 2.1 Initial Value Problem (IVP)

일반적인 Initial Value Problem을 다음과 같이 정의한다. 상태변수에 대한 초기값은 (2.1)와 같으며, 상태변수에 대한 미분방정식은 식(2.2)와 같다.

$$x(t_0) = x_0 \quad (2.1)$$

$$\dot{x} = f(t, x) \quad (2.2)$$

위 IVP에 대한 해는 시간 구간  $t \in [t_0, t_f]$ 에 대하여, 상태변수  $x(t)$ 를 구하는 문제이다. 위 해는 비선형 미분방정식과 같이 해석적인 닫힌해가 존재하지 않는 경우에는 수치 적분을 통하여 해를 계산한다. 대표적으로 Fixed-step 또는 Variable-step Runge-Kutta(RK) 방법이 있다. RK 적분방법의 장점은 적분의 강건성과 확장의 용이성 등을 들 수 있다.

### 2.2 IVP 문제의 해

#### 2.2.1 Fixed Point Iteration (FPI)

Fixed Point Iteration 방법은 IVP의 해를 근사

화하여 계산하는 방법으로, RK 방법과는 다른 방식으로 적용된다. FPI 방법은 다음과 같이 설명할 수 있다.  $x(t)$ 의 일반적인 해는 다음과 같이 식(2.2)로 나타낼 수 있다.

$$x(t) = x(t_0) + \int_{t_0}^{t_f} \dot{x}(s) ds \quad (2.3)$$

식(2.2)를 식(2.3)에 대입하면, 식(2.3)은 아래와 같이 나타낼 수 있다.

$$x(t) = x(t_0) + \int_{t_0}^{t_f} f(s, x(s)) ds \quad (2.4)$$

식 (2.4)의 왼쪽 및 오른쪽 항에 있는  $x(t)$ 의 근사화된 함수를 각각  $\phi_N(t)$ ,  $\phi_{N+1}(t)$ 로 나타내면 다음과 같은 식으로 나타낼 수 있다.

$$\phi_{N+1}(t) = x(t_0) + \int_{t_0}^{t_f} f(s, \phi_N(s)) ds \quad (2.5)$$

식(2.5)를 연속적으로 반복하여 계산을 수행하게 되면,  $x(t)$ 는  $\phi_\infty(t)$ 로 근사화할 수 있다. 위 방법에 대한 간단한 예를 다음과 같이 살펴보았다. 상태변수  $x$ 의 초기 값은 1로 설정하고, 상태변수에 대한 운동방정식은 간단히 다음과 같이 가정하였다.

$$\dot{x} = x \quad (2.6)$$

$\phi_0(t)$ 의 초기 근사화 값은 0으로 가정하고, 식 (2.5)를 계산하면, 다음과 같이 식(2.7)과 같이 계산된다.

$$\begin{aligned} \phi_1(t) &= x(t_0) + \int_0^t f(s, \phi_0(s)) ds \quad (2.7) \\ &= 1 + \int_0^t 0 ds = 1 \end{aligned}$$

위 과정을 N+1번 수행하면, 다음과 같이 식(2.8)과 같이 계산된다.

$$\phi_{N+1}(t) = 1 + t + \frac{1}{2}t^2 + \dots + \frac{1}{N!}t^N \quad (2.8)$$

한편, N을 무한 번 수행하면, 식 (2.8)은 다음과 같이 나타낼 수 있다.

$$\phi_\infty(t) = \sum_{k=0}^{\infty} \frac{t^k}{k!} \quad (2.9)$$

식 (2.9)는 예제의 해인  $e^t$  함수의 Taylor series

결과와 같은 것을 볼 수 있다. 미분방정식  $f$ 가 선형인 경우에는 식(2.5)의 오른쪽항의 적분을 해석적으로 계산할 수 있다. 하지만, 미분방정식  $f$ 가 비선형 식인 경우에는 식(2.5)의 오른쪽 항에 나타난 적분을 해석적으로 계산하기 어렵거나 불가능하기 때문에, 위 적분을 근사화하는 방법을 적용한다.

## 2.2.2 Numerical Fixed Point Iteration (NFPI)

비선형 미분방정식을 적용하기 위해서는 식 (2.5)의 오른쪽 항에 있는 적분을 근사화하여 적용한다. 시간 구간  $t \in [t_0, t_f]$ 에 대하여,  $\dot{x}(t)$ 를 근사적으로 적분하는 방법으로는 여러 방법이 있으며, 본 연구에서는 Quadrature rule을 소개한다. Quadrature rule은 특정 범위에서 정의된 함수  $\dot{x}(t)$ 의 적분을 근사화하는 방법으로, 범위내의 상태변수와 가중치(weight)의 조합으로 다음과 같이 표현된다.

$$\int_{t_0}^{t_f} \dot{x}(t) dt = \sum_{i=0}^n w_i f(x(t_i)) \quad (2.10)$$

위 식에서  $w_i, x(t_i)$ 는 각각 시간  $t_i$ 에서의 가중치와 상태변수를 나타낸다. 따라서, 비선형 미분방정식의 N-th iteration에서는 해당하는 시점에서의 상태변수를 이용하여, 함수값  $f$ 를 계산하고, 가중치를 적용하여, 적분된 값을 계산하게 된다. 이때 적분에 사용되는 시간  $t_i$ 와 가중치  $w_i$ 는 방법에 따라 여러 가지 종류가 있으며, 본 연구에서는 Legendre-Gauss-Lobatto(LGL) point를 적분에 적용하기 위한 시점으로 이용하였다.

Legendre 미분방정식(differential equation)을 만족하는 다항식 해들에 대하여, 상호 직교하도록 구성하고, 이들을 낮은 차수부터 구성된 다항식의 집합을 Legendre 다항식이라 할 수 있으며,  $\tau \in [-1, 1]$ 에서 정의된다.  $L_N$ 을 N차의 Legendre 다항식이라 하면, 다음과 같은 관계식을 만족한다.

$$\begin{aligned} (N+1)L_{N+1}(\tau) \\ = (2N+1)\tau L_N(\tau) - NL_{N-1}(\tau) \end{aligned} \quad (2.11)$$

그리고,  $L(\tau)_0 = 1, L_1(\tau) = \tau$  로 정의할 수 있다. LGL point는 -1과 1을 포함하고  $L_N$ 의  $\tau$ 에 대한 미분값이 '0'이 되는 지점으로 정의된다. 위 지점은 수치적으로 root-finding 기법을 이용하여 계산되거나, modified tri-diagonal Jacobi matrix의 eigenvalue를 찾음으로써 보다 정교한 지점을 계산할 수 있다[3].

위 LGL point는 -1에서 1사이에서 정의되는 값으로, 실제 시간 영역과의 mapping은 다음과 같은 관계식으로 나타낼 수 있다.

$$t = \frac{[t_f - t_0]\tau + [t_f + t_0]}{2} \quad (2.12)$$

LGL point에서 정의된 벡터를  $\mathbf{x}^N$ 이라고 정의하면,  $\mathbf{x}^N$ 에서의 적분값은  $\mathbf{X}_N$ 은 다음과 같이 나타낼 수 있다.

$$\mathbf{X}_N = \frac{t_f - t_0}{2} \mathbf{I} \mathbf{x}_N + \mathbf{x}_0 \quad (2.13)$$

위 식에서  $\mathbf{x}_0$ 는 상태변수의 초기 값으로만 나타내어진 벡터이며,  $\mathbf{I}$ 은 적분행렬로서 다음과 같이 나타낼 수 있다.

$$\mathbf{I}_{i-1,k} = \frac{w_k}{2} \{1 + \tau_i\} + \frac{w_k}{2} \left\{ \sum_{j=1}^N L_j(\tau_k) [L_{j+1}(\tau_i) - L_{j-1}(\tau_i)] \right\} \quad (2.14)$$

### 3. Numerical example

#### 3.1 Equation of motion

수치예제에 이용되는 운동방정식은 궤도운동 방정식으로 지구, 달, 태양의 중력장 모델이 포함되었으며, 지구 중력장에는 J2-J4의 섭동력이 포함되었다. 상태변수  $\mathbf{x} = [x, y, z, v_x, v_y, v_z]^T$ 는 지구중심의 관성좌표계의 직교좌표계에서 나타낸 위치 및 속도 벡터의 성분이다. 위 좌표계에서의 운동방정식은 다음과 같다.

$$\dot{x} = v_x \quad (3.1)$$

$$\dot{y} = v_y \quad (3.2)$$

$$\dot{z} = v_z \quad (3.3)$$

$$\dot{v}_x = -\frac{\mu_{Earth}}{\|r\|^3} x + f_x^{J2,3,4} + f_x^{Sun} + f_x^{Moon} \quad (3.4)$$

$$\dot{v}_y = -\frac{\mu_{Earth}}{\|r\|^3} y + f_y^{J2,3,4} + f_y^{Sun} + f_y^{Moon} \quad (3.5)$$

$$\dot{v}_z = -\frac{\mu_{Earth}}{\|r\|^3} z + f_z^{J2,3,4} + f_z^{Sun} + f_z^{Moon} \quad (3.6)$$

위 식에서  $\|r\|$ 은 지구 중심에서 우주비행체까지의 거리를 의미하며,  $\mu_{Earth}$  지구 중력 상수,  $f_i^{Sun}, f_i^{Moon}, i = x, y, z$ 는 태양과 달이 우주비행체 운동에 끼치는 가속도의 각 성분을 나타낸다. 그리고  $f_i^{J2,3,4}, i = x, y, z$ 는 지구 중력의 J2, J3, J4에 대한 가속도를 각 성분으로 나타낸 항이다. 태양과 달의 중력에 대한 힘의 성분  $\vec{f}^t$ 의 구성은  $[f_x^t, f_y^t, f_z^t]$ 로서 윗첨자 t는 태양 또는 달을 의미한다.

$$\vec{f}^t = \mu_t \frac{\vec{r}_{s2t}}{\|\vec{r}_{s2t}\|^3} - \mu_t \frac{\vec{r}_t}{\|\vec{r}_t\|^3} \quad (3.7)$$

여기에서  $\mu_t$ 는 태양 또는 달의 중력상수를 의미하며,  $\vec{r}_{s2t}$ 는 우주비행체로부터 태양 또는 달까지의 거리,  $\vec{r}_t$  지구 중심으로부터 태양 또는 달까지의 거리를 나타낸다.

#### 3.2 Modified NFPI

수치 적분을 위한 Numerical fixed point iteration 기법을 본 예제에 적용하기 위하여 다음과 같은 순서를 적용하였다.

1) 먼저, DE405 또는 DE421과 같은 천체력으로부터 태양과 달의 위치를 계산하기 위하여, 시작 시간, 적분 구간, 그리고 Legendre 다항식의 차수N을 입력받아, LGL point에 해당하는 시간을 식 (2.12)로부터 계산한다.

2) LGL point에 해당하는 시간에 대하여, Julian date으로 환산한 이후, 태양과 달의 위치를 계산한다.

3) 각 상태변수의 LGL point에 해당하는 초기 벡터를 각 상태변수의 초기값으로 설정한다. 계산된 태양과 달의 위치 정보와 식(3.1)-(3.7)을 식(2.13)에 대입하여, iteration을 수행한다.

4) k번째 iteration에 대한, 최종시간에서의 적분된 값을 저장하고, k+1번째의 iteration을 수행한다.

5) 최종시간에서의 k+1번째와 k번째의 적분된 값의 차이  $e_{k+1}$  를 저장하고, 다음 iteration에서도 k+2번째와 k+1번째의 적분된 값의 차이  $e_{k+2}$ 를 계산한 다음,  $e_{k+2}$ 과  $e_{k+1}$ 의 차이가 사용자가 지정한 범위 이내에 도달하면, iteration을 종료한다.

6) 한편, Legendre 다항식의 차수를 사용자가 지정한 간격으로 증가시킨 후 2)-5)까지 일련의 계산 과정을 수행한다.

7) 두 Legendre 다항식 차수에 대한 최종 적분 값의 오차가 사용자가 설정한 범위 이내로 들어올 때 까지 반복한다.

### 3.3 수치 예제

수치 예제를 위해 고도 300km의 지구 중심원 궤도에서 고도 100km의 달 중심 원궤도까지 5일에 거쳐 천이하기 위한 궤적에 대하여 0.5일 시점의 위치 및 속도 데이터를 본 예제의 초기값으로 설정하였다. 5일에 대한 천이궤적은 그림 1에 나타나 있다. 표 1은 3.2절에서 서술된 방법을 적용하여 수치적분을 수행한 결과이다. Legendre 다항식의 차수는 20에서 시작하여, 25, 30, 40까지 적용된 것을 볼 수 있다. 최종시간에서의 오차는 Runge-Kutta방법을 이용한 결과와 비교하였으며, matlab의 ode45함수를 이용하였다. 최종시간에서의 오차는 약 1.82km에서 감소하여  $10^{-3}$  이하로 감소하는 것을 볼 수 있다. 계산 시간은 20차에 대하여 0.0023초가 소요되었으며, 40차에 대해서는 0.0033초가 소요된 것을 볼 수 있다. 그림 2는 Legendre 다항식이 30차인 경우, iteration 횟수가 증가함에 따라 거리 오차의 경향을 살펴 보았다. 약 10번째 iteration 이후에는 오차의 변화가 거의 없는 것을 확인할 수 있다.

성능비교를 위하여, 수치적분을 기존의 RK 방법과 비교해 보았다. Matlab의 내장함수 @ode45를 이용한 방법은 9.2초가 소요되었으며, 30초의 적분간격을 고정하여 fixed-step, fourth-order RK방법을 적용하였을 경우에는 69.21초가 소요되었다.

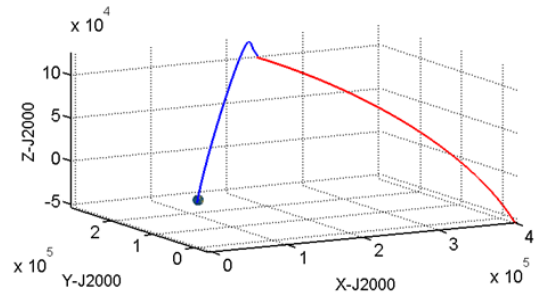


그림 1. 5-day Earth-to-Moon transfer trajectory (From 300-km Low Earth Orbit to 100-km Low Lunar Orbit)

표 1. 차수에 따른 적분 에러 및 계산 시간

차수	오차 (km)	계산시간 (sec)
20	1.82	0.0023
25	0.09	0.0025
30	0.004	0.0028
40	0.001	0.0033

위 경우에는 Variable-step을 적용하였을 경우와 비교하여 3.632km의 거리 오차를 나타내었다. 본 연구에서 진행되는 수치 적분결과와 비교하여 보면, 본 연구의 수치적분 방식이 계산 시간적인 면에서 월등히 나은 성능을 보이고 있으며, 정밀도 면에서도 높은 정밀도를 나타내고 있는 것을 볼 수 있다. 계산 시간의 월등한 차이를 나타내는 이유는 적분이 수행된 step을 비교하면 차이를 짐작할 수 있다. Variable-step을 이용한 @ode45의 경우 총 2327횟수의 적분 step을 나타내고 있으며, 30초 등간격의 Fixed-step RK4차의 경우에는 12960횟수의 적분 step을 나타내고 있

다. 3.2절에서의 방식으로는 40차의 Legendre 다항식의 경우 41개의 지점에서만 적분이 계산되고 근사화되기 때문에 보다 효과적으로 적분이 이루어지는 것을 확인할 수 있다.

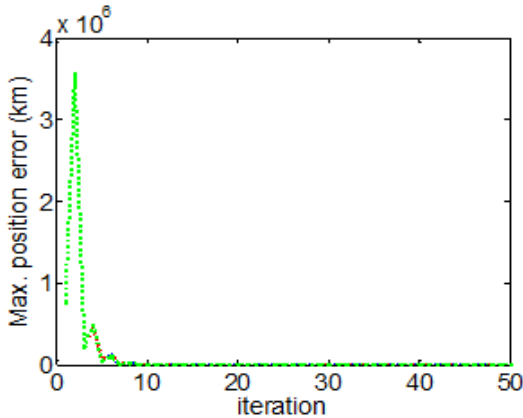


그림 2. Legendre 다항식 30차의 경우, iteration에 따른 오차 경향

표 2. RK 방법을 이용한 수치적분 결과

방법	계산시간 (sec)
ode45	9.20
Fixed step RK4 method (step 30sec)	69.21 (오차 : 3.632 km)

## 4. 결 론

본 연구에서는 Fixed Point Iteration 기법을 적용한, 비선형 운동방정식의 수치 적분 문제를 다루었다. 이를 위해, Quadrature rule이 적용된 적분행렬을 이용하여, 비선형 운동방정식에 대한 초기값 문제(Initial values problem)를 다루었다. 수치 예제를 통하여, 지구, 달, 태양의 중력이 고려된 궤도운동에 대한 수치 적분 시간이 본 알고리즘을 통하여 0.01초 소요되는 것을 확인할 수 있었다. 이는 ODE45를 사용한 경우의 9.2초가 소요된 것을 비교하면, 계산 시간적 측면에서 많은 성능 향상이 이루어진 것을 확인할 수 있었

다. Quadrature rule에 사용된 Legendre polynomial의 차수에 따라 적분 정밀도가 달라지는데, 오차가 사용자가 지정한 범위 이내로 들어올 수 있도록, 스스로 차수를 변경할 수 있는 알고리즘을 소개하였다.

## 참 고 문 헌

1. John T. Betts, "Survey of Numerical Methods for Trajectory Optimization," Journal of Guidance, Control, and Dynamics, Vol.21, 1998, pp. 193-207.
2. Bai, X., and Junkins, J.L., "Modified Chebyshev-Picard Iteration Methods for solution of Boundary Value Problem," Journal of the Astronautical Sciences, Vol. 58, No.4, 2011, pp. 615-642.
3. Gautschi, W., "High-order Gauss-Lobatto formulae" Numerical Algorithm, Vol. 25, 2000, PP. 213-222.