NLOptControl Documentation

Release 0.0.1-rc1

Huckleberry Febbo

Apr 08, 2017

Contents

CHAPTER 1

Table of Contents

Background Information

While detailed information on these approaches to discretizing infinite dimensional (or continuous) optimal control problems can be found (and comes from) this Ph.D. dissertation, this related journal publication and this technical report, the Background Information section will cover some basics.

Lagrange Interpolating Polynomials

Definition

- given $(N + 1)$ unique data points
  - $(x_0, y_0), (x_1, y_1), ..., (x_N, y_N)$
- we can create an N^{th} order Lagrange interpolating polynomial

$$P_n(x) = \sum_{i=0}^{N} \mathcal{L}_i(x) f(x_i)$$

where,

$$f(x_0) = y_0 \tag{1.1}$$
$$f(x_1) = y_1 \tag{1.2}$$
$$. \tag{1.3}$$
$$. \tag{1.4}$$
$$f(x_i) = y_i \tag{1.5}$$
$$. \tag{1.6}$$
$$f(x_N) = y_N \tag{1.7}$$

So, we are just multiplying by the given y_i values.

Lagrange Basis Polynomials

More information on Lagrange Basis Polynomials is here

$$\mathcal{L}_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{N} \frac{x - x_j}{x_i - x_j}$$

so expanding this,

$$\begin{aligned}
\mathcal{L}_i(x) &= \tfrac{x-x_0}{x_i-x_0}\tfrac{x-x_1}{x_i-x_1}\cdots && (1.8)\\
&\cdots\tfrac{x-x_{i-1}}{x_i-x_{i-1}}\cdots && (1.9)\\
&\cdots\tfrac{x-x_{i+1}}{x_i-x_{i+1}}\cdots && (1.10)\\
&\cdots\tfrac{x-x_N}{x_i-x_N} && (1.11)
\end{aligned}$$

Notice that we do not include the term where $i == j$!

Please see lpf for details on implementation.

Direct Transcription of Optimal Control Problems

Let $N_t + 1$ be the total number of discrete time points.

Time Marching Methods

Euler Method

Trapezoidal Method

Pseudospectral Methods

Change of Interval

To can change the limits of the integration (in order to apply Quadrature), we introduce $\tau \in [-1, +1]$ as a new independent variable and perform a change of variable for t in terms of τ, by defining:

$$\tau = \frac{2}{t_{N_t} - t_0}t - \frac{t_{N_t} + t_0}{t_{N_t} - t_0}$$

Polynomial Interpolation

Select a set of $N_t + 1$ node points:

$$\tau = [\tau_0, \tau_1, \tau_2, ..., \tau_{N_t}]$$

- These none points are just numbers
  - Increasing and distinct numbers $\in [-1, +1]$

A *unique* polynomial $P(\tau)$ exists (i.e. $\exists! P(\tau)$) of a maximum degree of N_t where:

$$f(\tau_k) = P(\tau_k), \quad k = 0, 1, 2, \ldots.N_t$$

- So, the function evaluated at τ_k is equivalent the this polynomial evaluated at that point.

But, between the intervals, we must approximate $f(\tau)$ as:

$$f(\tau) \approx P(\tau) = \sum_{k=0}^{N_t} f(\tau_k)\phi_k(\tau)$$

with $\phi_k()$ are basis polynomials that are built by interpolating $f(\tau)$ at the node points.

Approximating Derivatives

We can also approximate the derivative of a function $f(\tau)$ as:

$$\frac{\mathrm{d}f(\tau)}{\mathrm{d}\tau} = \dot{f}(\tau_k) \approx \dot{P}(\tau_k) = \sum_{i=0}^{N_t} D_{ki} f(\tau_i)$$

With $\mathbf{D}$ is a $(N_t + 1) \times (N_t + 1)$ differentiation matrix that depends on:

- values of τ
- type of interpolating polynomial

Now we have an approximation of $\dot{f}(\tau_k)$ that depends only on $f(\tau)$!

Approximating Integrals

The integral we are interested in evaluating is:

$$\int_{t_0}^{t_{N_t}} f(t)\mathrm{d}t = \frac{t_{N_t} - t_0}{2} \int_{-1}^{1} f(\tau_k)\mathrm{d}\tau$$

This can be approximated using quadrature:

$$\int_{-1}^{1} f(\tau_k)\mathrm{d}\tau \sum_{k=0}^{N_t} w_k f(\tau_k)$$

where w_k are quadrature weights and depend only on:

- values of τ
- type of interpolating polynomial

Legendre Pseudospectral Method

- Polynomial

Define an N order Legendre polynomial as:

$$L_N(\tau) = \frac{1}{2^N N!}\frac{\mathrm{d}^n}{\mathrm{d}\tau^N}(\tau^2 - 1)^N$$

- Nodes

$$\tau_k = \begin{cases} -1, & \text{if } k = 0 \\ \text{kth root } of \dot{L}_{N_t}(\tau), & \text{if } k = 1,2,3,..N_t - 1 \\ +1 & \text{if } k = N_t \end{cases} \tag{1.12}$$

- Differentiation Matrix
- Interpolating Polynomial Function

hp-psuedospectral method

To solve the integral constraints within the optimal control problem we employs the hp-pseudospectral method. The hp-psuedospectral method is an form of Gaussian Quadrature, which uses multi-interval collocation points.

Single Phase Optimal Control

Find:

- The state: $\mathbf{x}(t)$
- The control: $\mathbf{u}(t)$
- The integrals: $\mathbf{q}$
- The initial time: t_0
- The final time: t_f

To Minimize:

$$J = \Phi(\mathbf{x}(t_0), \mathbf{x}(t_f), \mathbf{q}, t_0, t_f)$$

That Satisfy the Following Constraints:

- Dynamic Constraints:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \psi(\mathbf{x}(t), \mathbf{u}(t), t)$$

- Inequality Path Constraints:

$$\mathbf{c}_{min} <= \mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) <= \mathbf{c}_{max}$$

- Integral Constraints:

$$q_i = \int_{t_0}^{t_f} \Upsilon_i(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t, \quad (i = 1, ..., n_q)$$

- Event Constraints:

$$\mathbf{b}_{min} <= \mathbf{b}(\mathbf{x}(t_0), \mathbf{x}(t_f), t_f, \mathbf{q}) <= \mathbf{b}_{max}$$

Change of Interval

To can change the limits of the integration (in order to apply Quadrature), we introduce $\tau \in [-1, +1]$ as a new independent variable and perform a change of variable for t in terms of τ, by defining:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}$$

The optimal control problem defined above (TODO: figure out equation references), is now redefined in terms of τ as:

Find:

- The state: $\mathbf{x}(\tau)$
- The control: $\mathbf{u}(\tau)$
- The integrals: $\mathbf{q}$
- The initial time: t_0
- The final time: t_f

To Minimize:

$$J = \Phi(\mathbf{x}(-1), \mathbf{x}(+1), \mathbf{q}, t_0, t_f)$$

That Satisfy the Following Constraints:

- Dynamic Constraints:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\tau} = \frac{t_f - t_0}{2}\psi(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau, t_0, t_f)$$

- Inequality Path Constraints:

$$\mathbf{c}_{min} <= \mathbf{c}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau, t_0, t_f) <= \mathbf{c}_{max}$$

- Integral Constraints:

$$q_i = \frac{t_f - t_0}{2}\int_{-1}^{+1} \Upsilon_i(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau, t_0, t_f)\, \mathrm{d}\tau, \quad (i = 1, ..., n_q)$$

- Event Constraints:

$$\mathbf{b}_{min} <= \mathbf{b}(\mathbf{x}(-1), \mathbf{x}(+1), t_f, \mathbf{q}) <= \mathbf{b}_{max}$$

Divide The Interval $\tau \in [-1, +1]$

The interval $\tau \in [-1, +1]$ is now divided into a mesh of K mesh intervals as:

$$[T_{k-1}, T_k], k = 1, ..., T_K$$

with $(T_0, ..., T_K)$ being the mesh points; which satisfy:

$$-1 = T_0 < T_1 < T_2 < T_3 < .... < T_{K-1} < T_K = T_f = +1$$

Rewrite the Optimal Control Problem using the Mesh

Find:

- The state : $\mathbf{x}^{(k)}(\tau)$ **in mesh interval k**
- The control: $\mathbf{u}^{(k)}(\tau)$ **in mesh interval k**
- The integrals: $\mathbf{q}$
- The initial time: t_0
- The final time: t_f

To Minimize:

$$J = \Phi(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1), \mathbf{q}, t_0, t_f)$$

That Satisfy the Following Constraints:

- Dynamic Constraints:

$$\frac{\mathrm{d}\mathbf{x}^{(k)}(\tau^{(k)})}{\mathrm{d}\tau^{(k)}} = \frac{t_f - t_0}{2}\psi(\mathbf{x}^{(k)}(\tau^{(k)}), \mathbf{u}^{(k)}(\tau^{(k)}), \tau^{(k)}, t_0, t_f), \quad (k = 1, ..., K)$$

- Inequality Path Constraints:

$$\mathbf{c}_{min} <= \mathbf{c}(\mathbf{x}^{(k)}(\tau^{(k)}), \mathbf{u}^{(k)}(\tau^{(k)}), \tau^{(k)}, t_0, t_f) <= \mathbf{c}_{max}, \quad (k = 1, ..., K)$$

- Integral Constraints:

$$q_i = \frac{t_f - t_0}{2} \sum_{k=1}^{K} \int_{T_{k-1}}^{T_k} \Upsilon_i(\mathbf{x}^{(k)}(\tau^{(k)}), \mathbf{u}^{(k)}(\tau^{(k)}), \tau, t_0, t_f)\, \mathrm{d}\tau, \quad (i = 1, ..., n_q, k = 1, ..., K)$$

- Event Constraints:

$$\mathbf{b}_{min} <= \mathbf{b}(\mathbf{x}^{(1)}(-1), \mathbf{x}^{(K)}(+1), t_f, \mathbf{q}) <= \mathbf{b}_{max}$$

- State Continuity
  - Also, we must **now** constrain the state to be continuous at each interior mesh point $(T_1, ...T_{k-1})$ by enforcing:

$$\mathbf{y}^k(T_k) = \mathbf{y}^{k+1}(T_k)$$

Optimal Control Problem Approximation

The optimal control problem will now be approximated using the Radau Collocation Method as which follows the description provided by *[BGar11]*. In collocation methods, the state and control are discretized at particular points within the selected time interval. Once this is done the problem can be transcribed into a nonlinear programming problem (NLP) and solved using standard solvers for these types of problems, such as IPOPT or KNITRO.

For each mesh interval $k \in [1, .., K]$**:**

$$\mathbf{x}^{(k)}(\tau) \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} \frac{\mathrm{d}\mathcal{L}_j^k(\tau)}{\mathrm{d}\tau} \tag{1.13}$$

$$where, \tag{1.14}$$

$$\mathcal{L}_j^k(\tau) = \prod_{\substack{l=1\\ l\neq j}}^{N_k+1} \frac{\tau-\tau_l^{(k)}}{\tau_j^{(k)}-\tau_l^{(k)}} \tag{1.15}$$

$$and, \tag{1.16}$$

$$D_{ki} = \dot{\mathcal{L}}_i(\tau_k) = \frac{\mathrm{d}\mathcal{L}_j^k(\tau)}{\mathrm{d}\tau} \tag{1.17}$$

also,

- $\mathcal{L}_j^{(k)}(\tau), \quad (j = 1, ..., N_k + 1)$ is a basis of Lagrange polynomials
- $(\tau_1^k, ..., \tau_{N_k}^{(k)})$ are the Legendre-Gauss-Radau collocation points in mesh interval k
  - defined on the subinterval $\tau^{(k)} \in [T_{k-1}, T_k]$
  - $\tau_{N_k+1}^{(k)} = T_k$ is a noncollocated point

A basic description of Lagrange Polynomials is presented in *Lagrange Interpolating Polynomials*

The $\mathbf{D}$ **matrix:**

- **Has a size** $= [N_c] \times [N_c + 1]$
  - with $(1 <= k <= N_c), (1 <= i <= N_c + 1)$
  - **this non-square shape because the state approximation uses the** $N_c + 1$ **points:** $(\tau_1, ...\tau_{N_c+1})$
  - **but collocation is only done at the** N_c **LGR points:** $(\tau_1, ...\tau_{N_c})$

If we define the state matrix as:

$$\mathbf{X}^{LGR} = \begin{bmatrix} \mathbf{X}_1 \\ . \\ . \\ . \\ \mathbf{X}_{N_c+1} \end{bmatrix} \tag{1.18}$$

The dynamics are collocated at the N_c LGR points using:

$\mathbf{D}_k \mathbf{X}^{LGR} = \frac{(t_f - t_0)}{2} \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k, \tau, t_0, t_f) \quad for \quad k = 1, ..., Nc$

with,

- $\mathbf{D}_k$ being the k^{th} row of the $\mathbf{D}$ matrix.

References

Examples

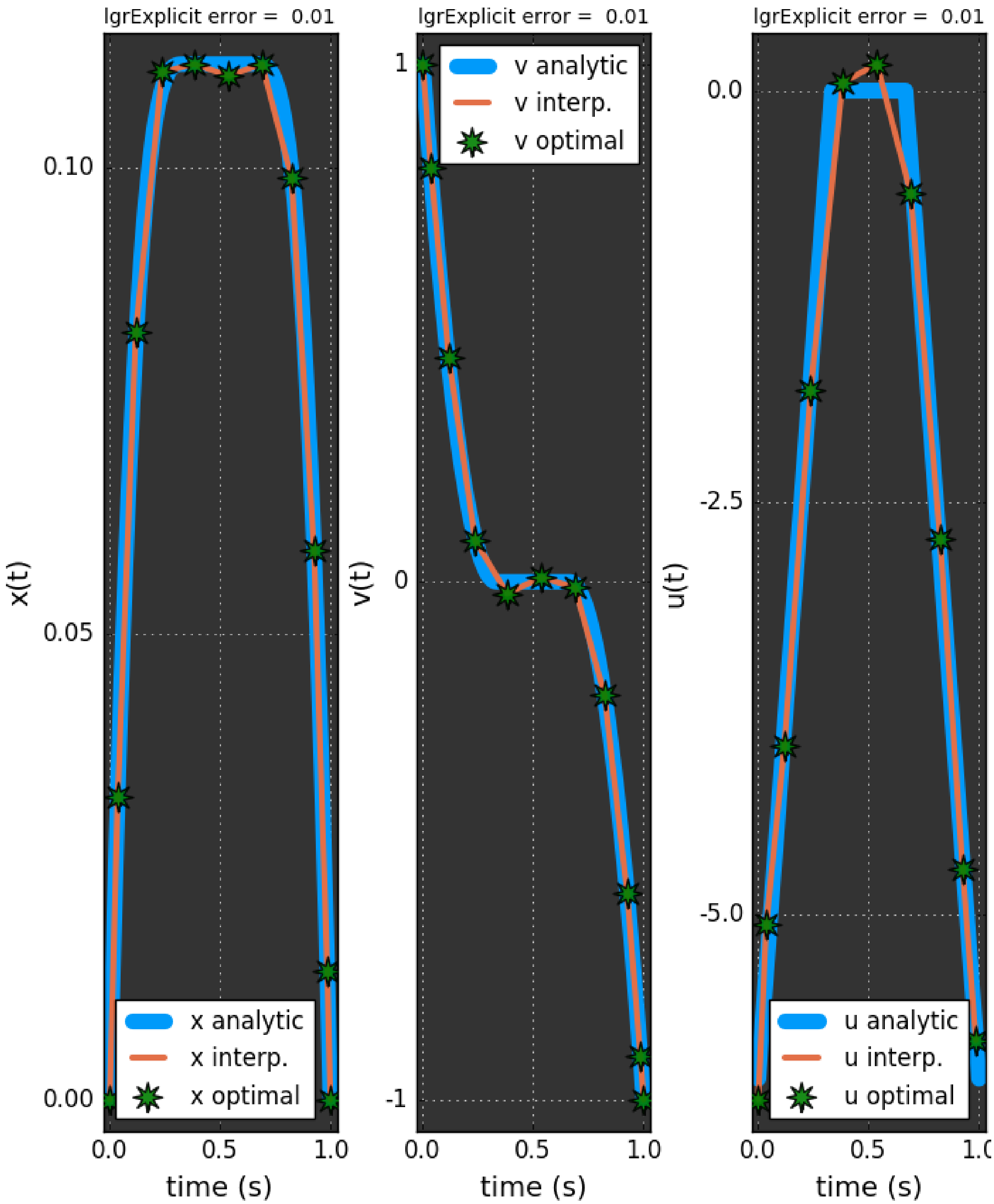
Optimal Control Problems

Bryson-Denham

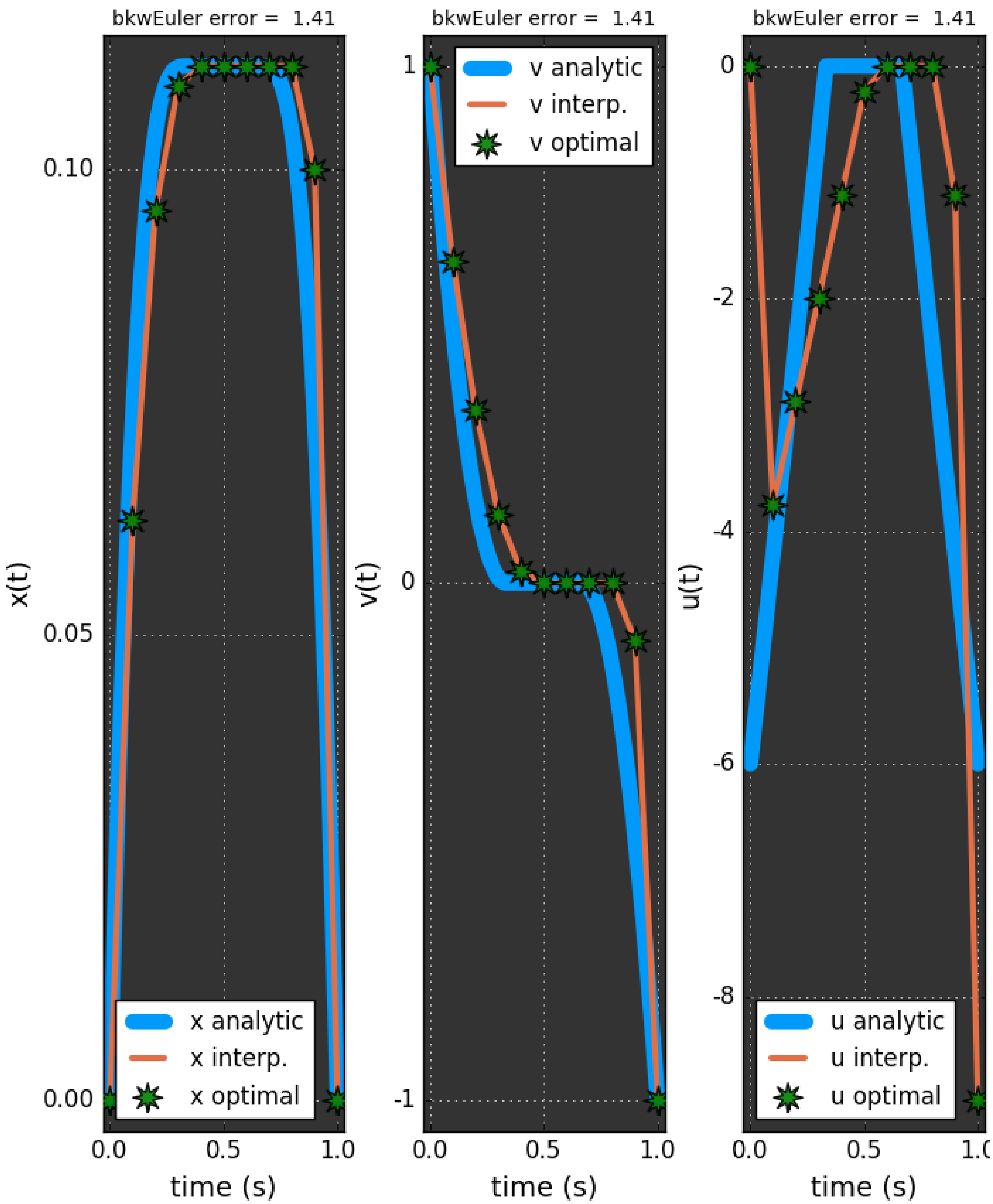
http://www.gpops2.com/Examples/Bryson-Denham.html

`N = 10` –> ex#1

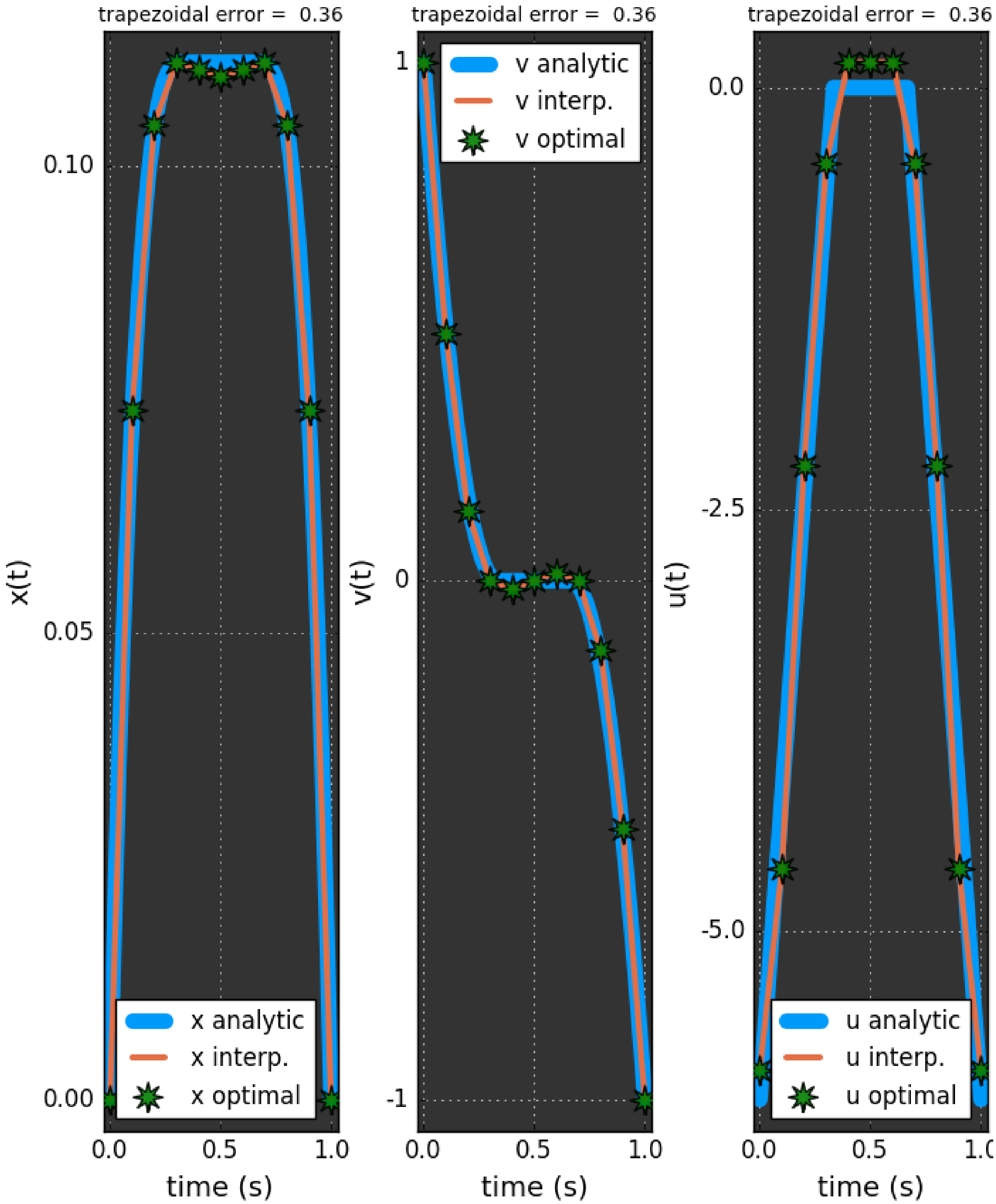
LGR


lgrExplicit error = 0.01
lgrExplicit error = 0.01
lgrExplicit error = 0.01
x(t)
v(t)
u(t)
0.10
0.05
0.00
1
0
-1
0.0
-2.5
-5.0
x analytic
x interp.
x optimal
v analytic
v interp.
v optimal
u analytic
u interp.
u optimal
0.0
0.5
1.0
time (s)
time (s)
time (s)

Bkw Euler

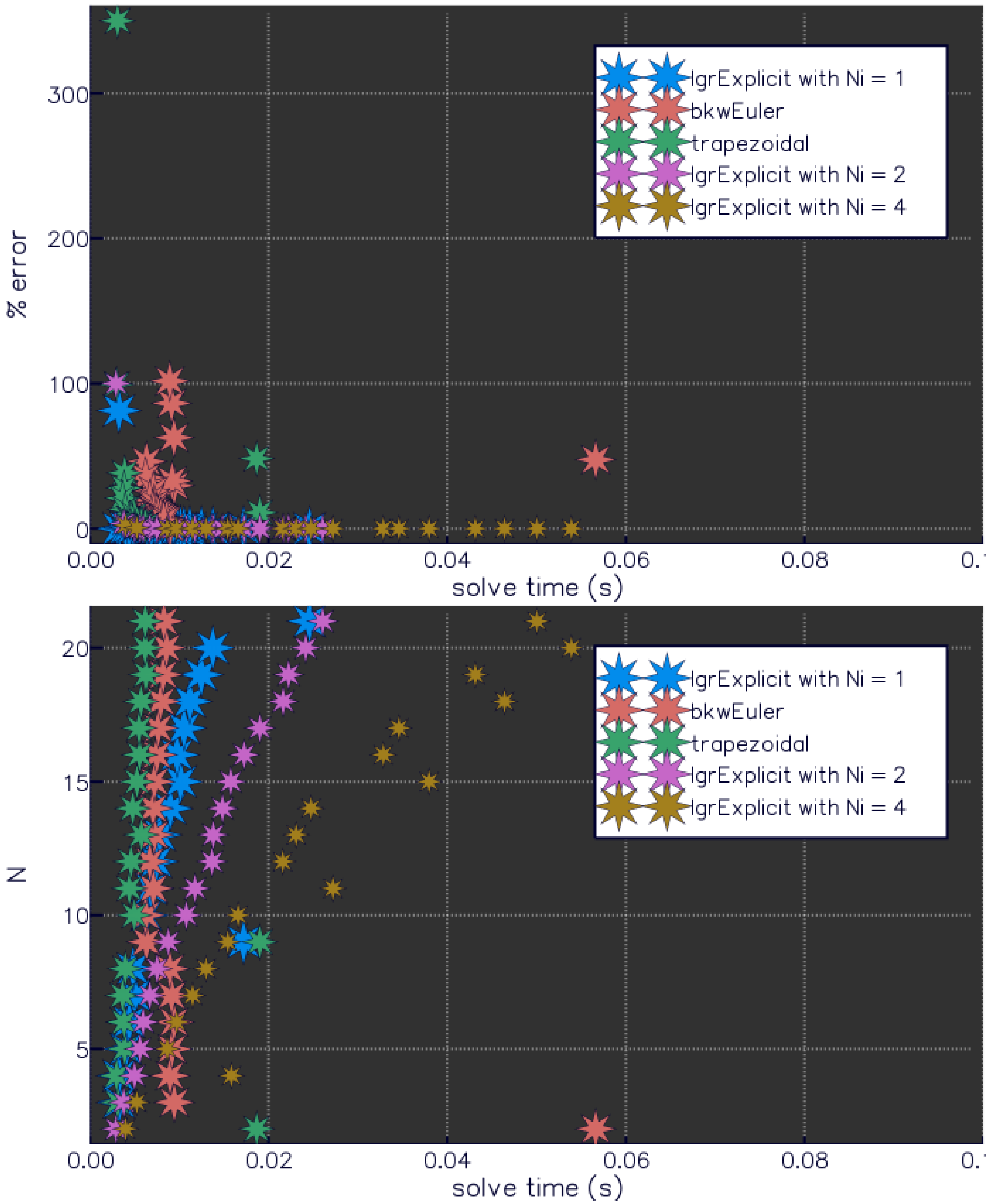

bkwEuler error = 1.41
bkwEuler error = 1.41
bkwEuler error = 1.41
x(t)
v(t)
u(t)
0.10
0.05
0.00
1
0
-1
0
-2
-4
-6
-8
x analytic
x interp.
x optimal
v analytic
v interp.
v optimal
u analytic
u interp.
u optimal
0.0
0.5
1.0
time (s)
time (s)
time (s)

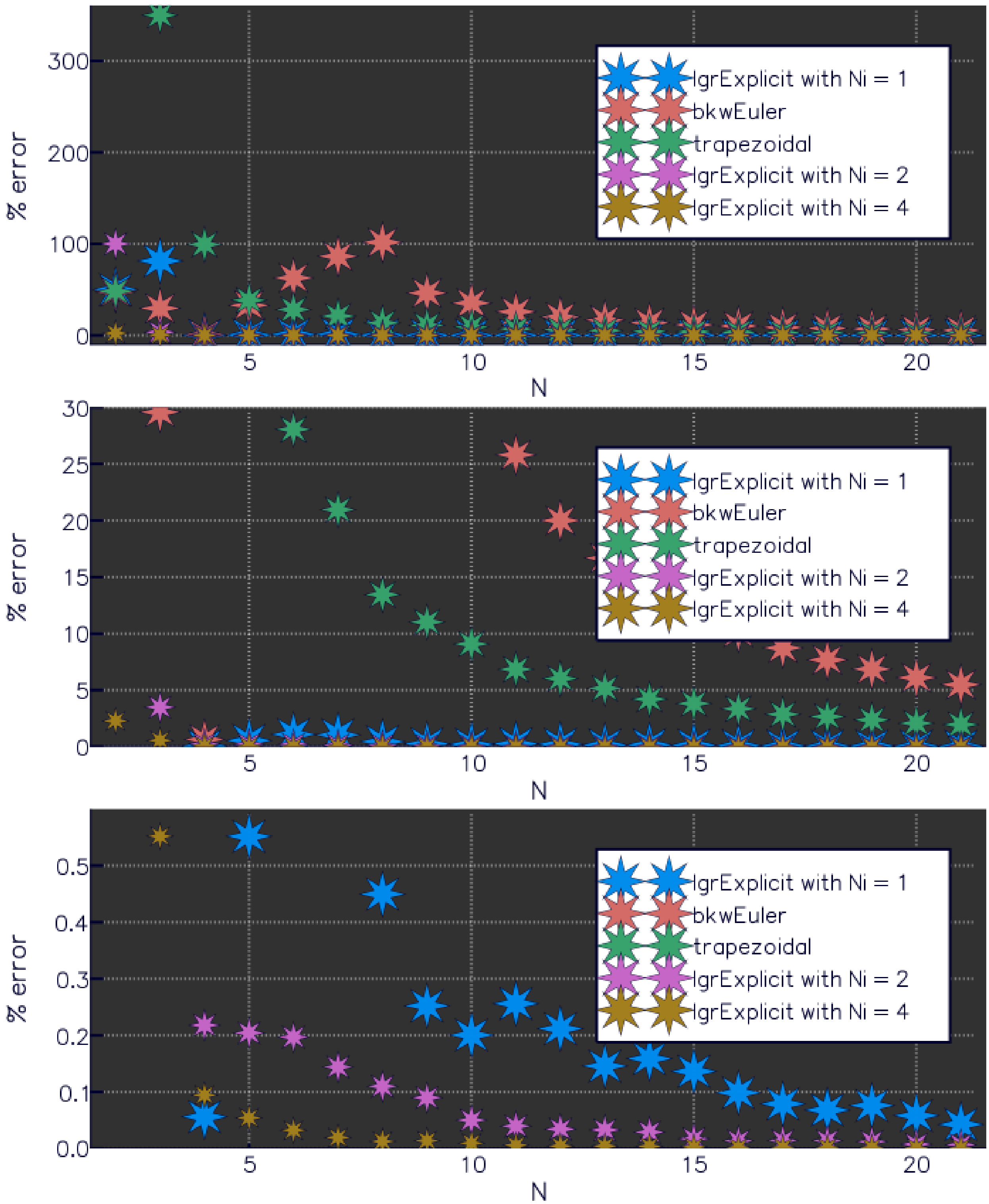
Trapezoidal



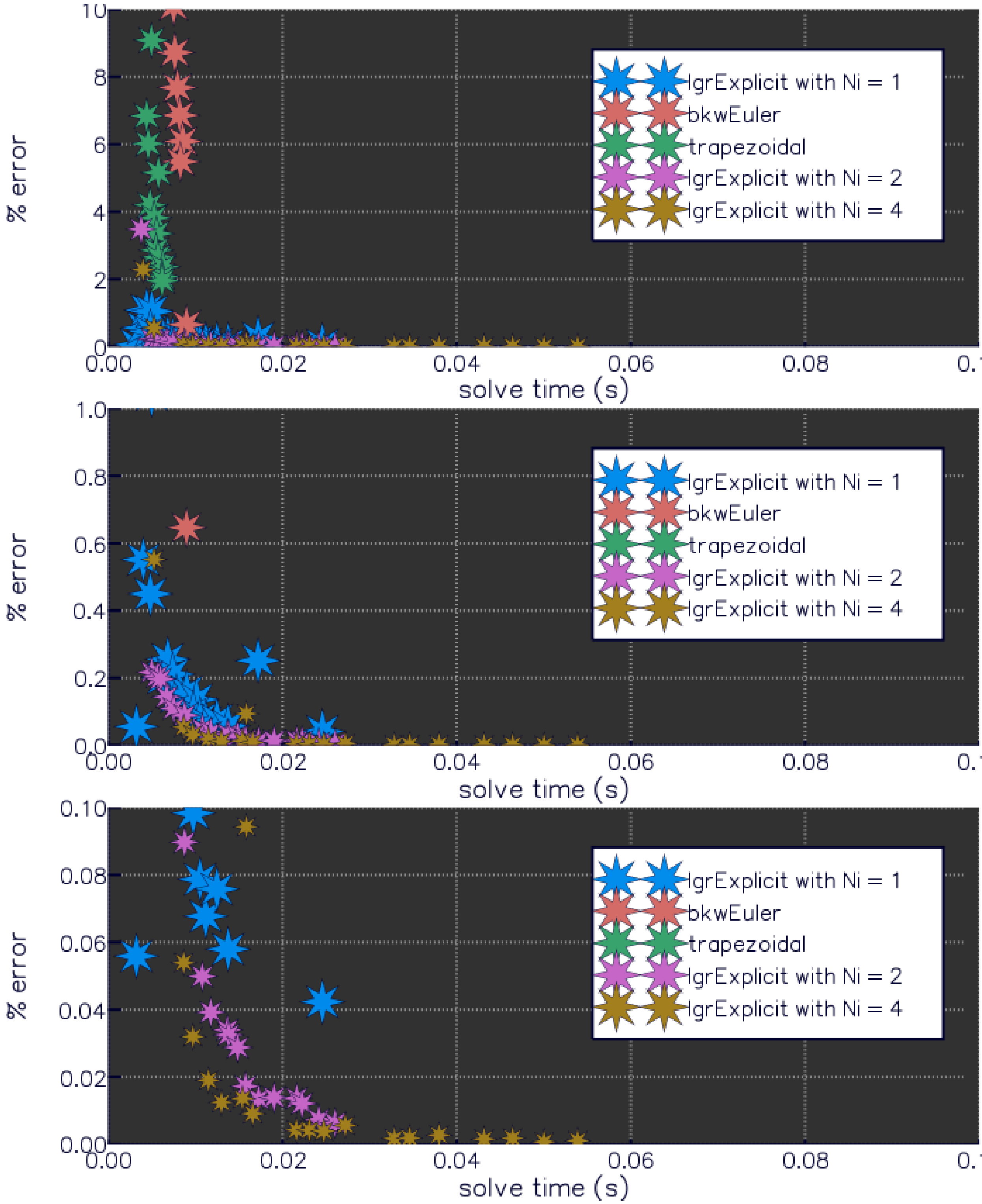
Error vs. Speed –> ex#5

Initial Solve

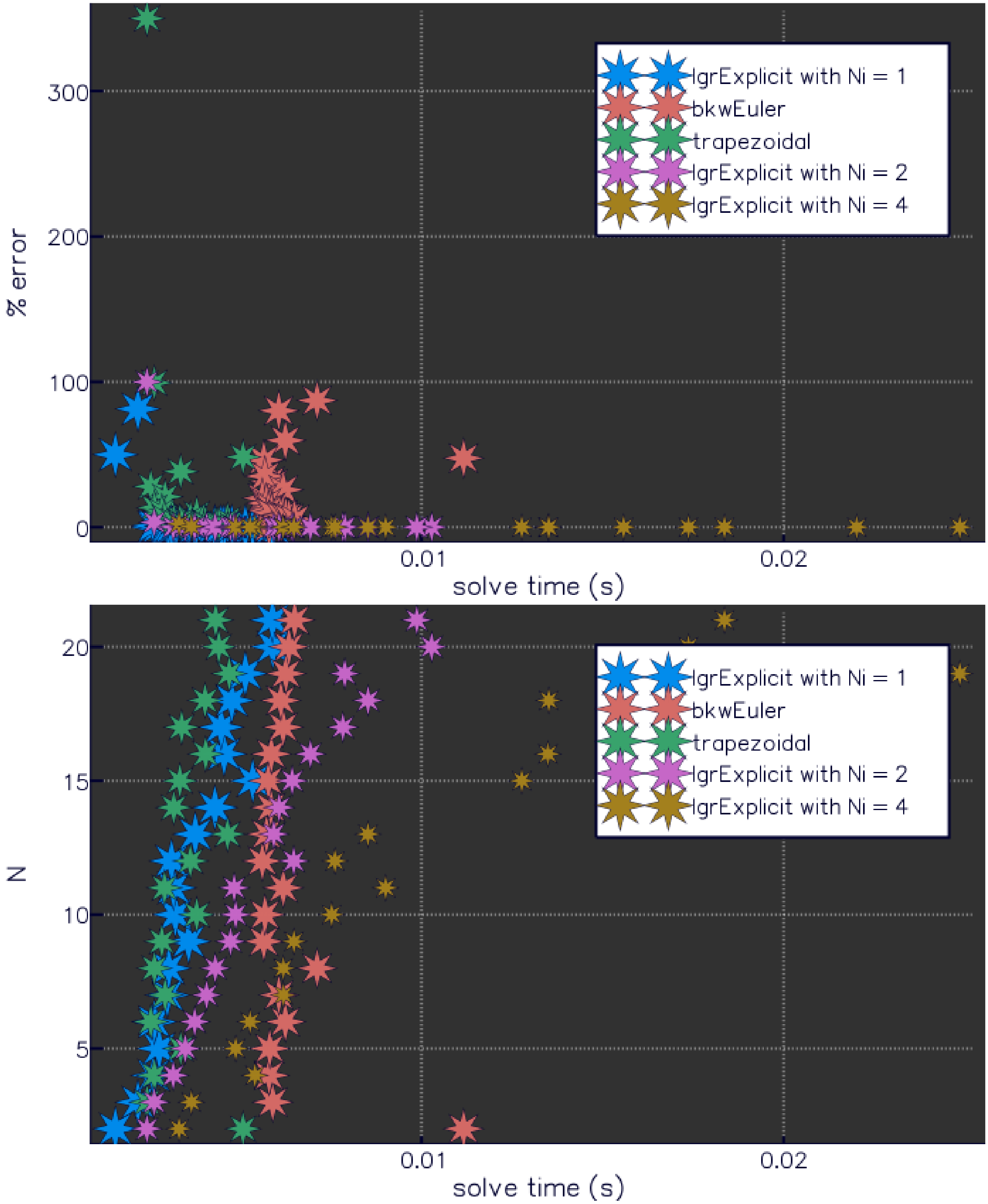


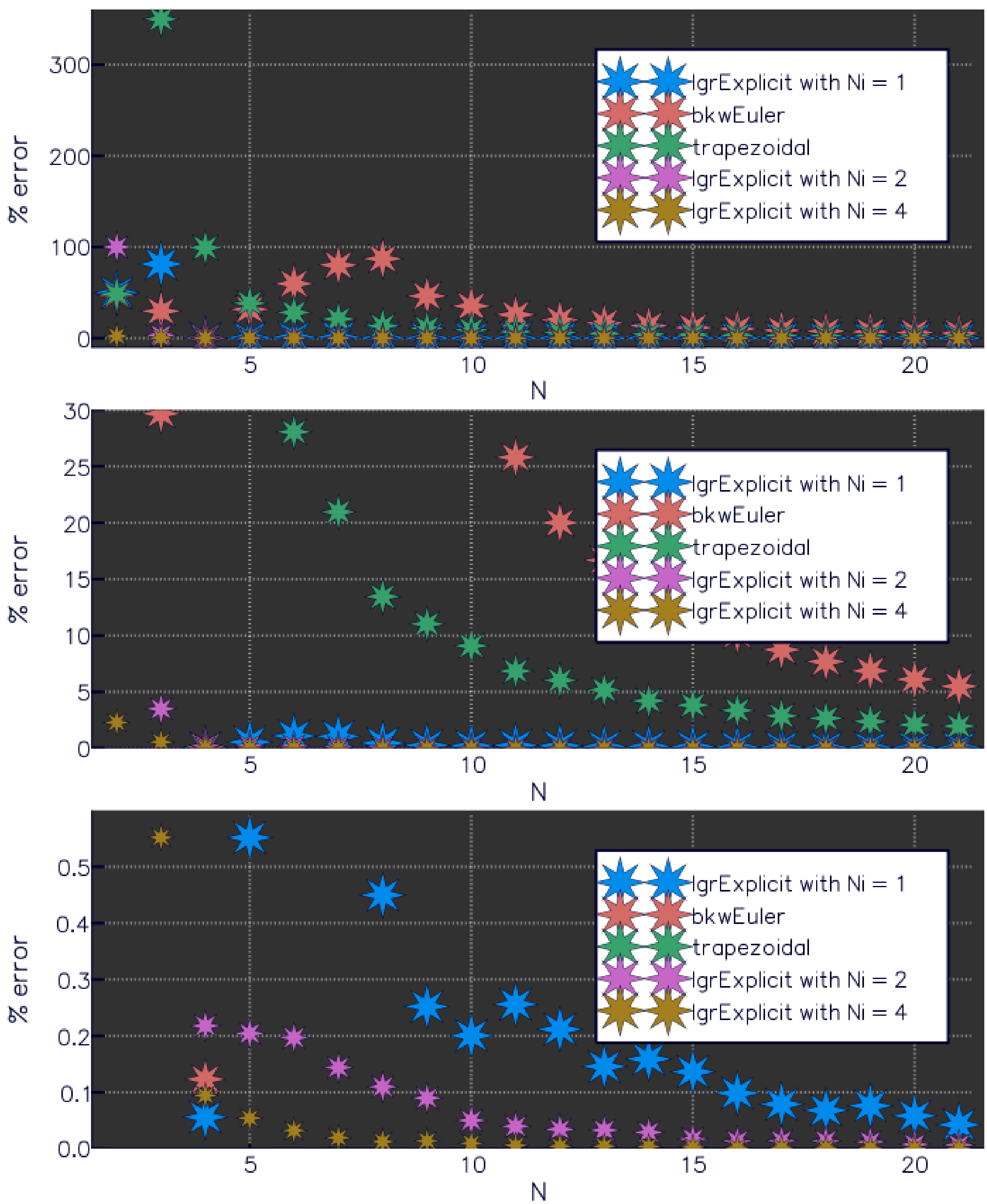


300
200
100
0
% error
5
10
15
20
N
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4
30
25
20
15
10
5
0
% error
5
10
15
20
N
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4
0.5
0.4
0.3
0.2
0.1
0.0
% error
5
10
15
20
N
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4

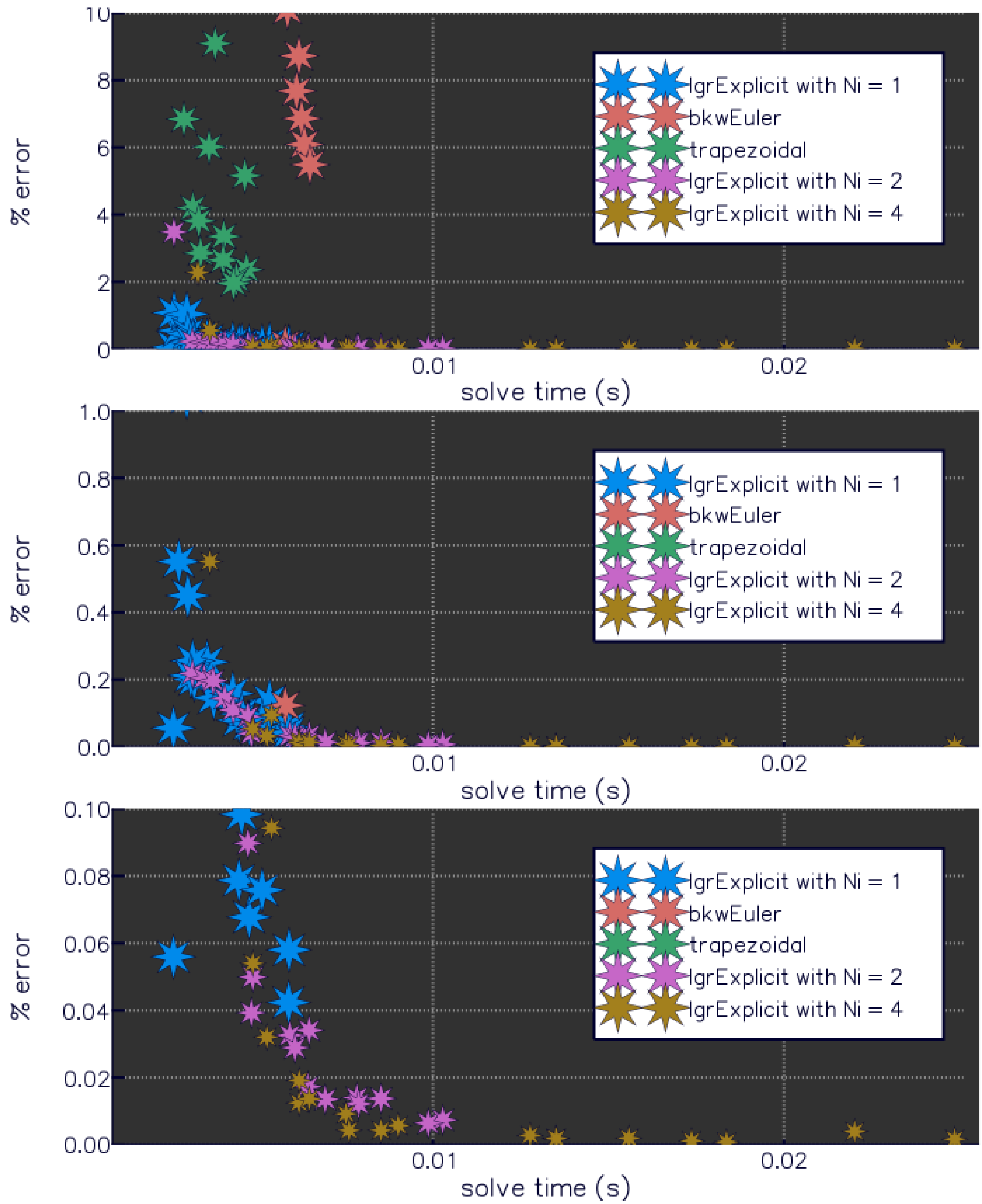

% error
solve time (s)
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4
% error
solve time (s)
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4
% error
solve time (s)
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4

Second Solve





% error
N
lgrExplicit with Ni = 1
bkwEuler
trapezoidal
lgrExplicit with Ni = 2
lgrExplicit with Ni = 4

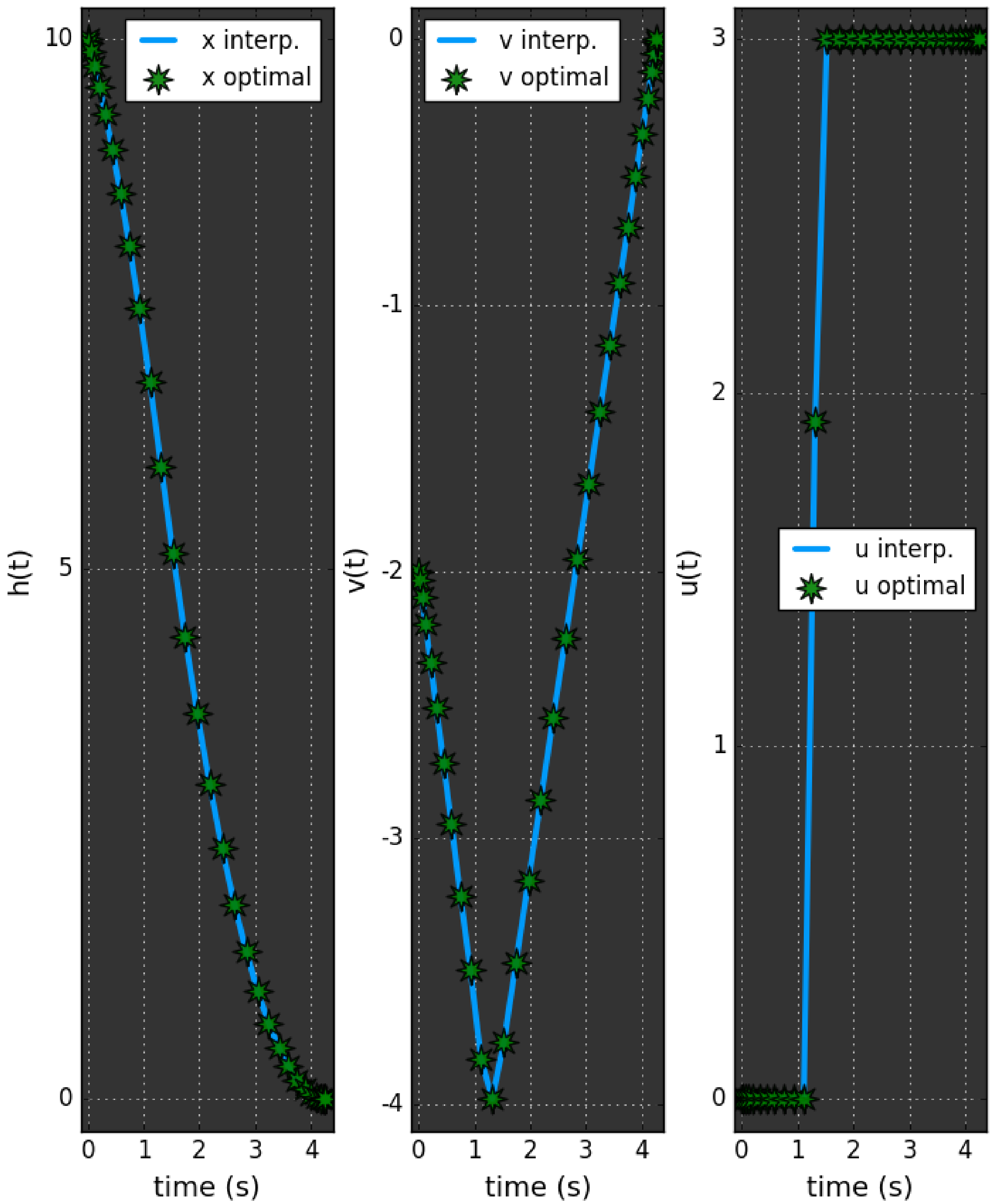


Moon Lander

http://gpops2.com/Examples/MoonLander.html

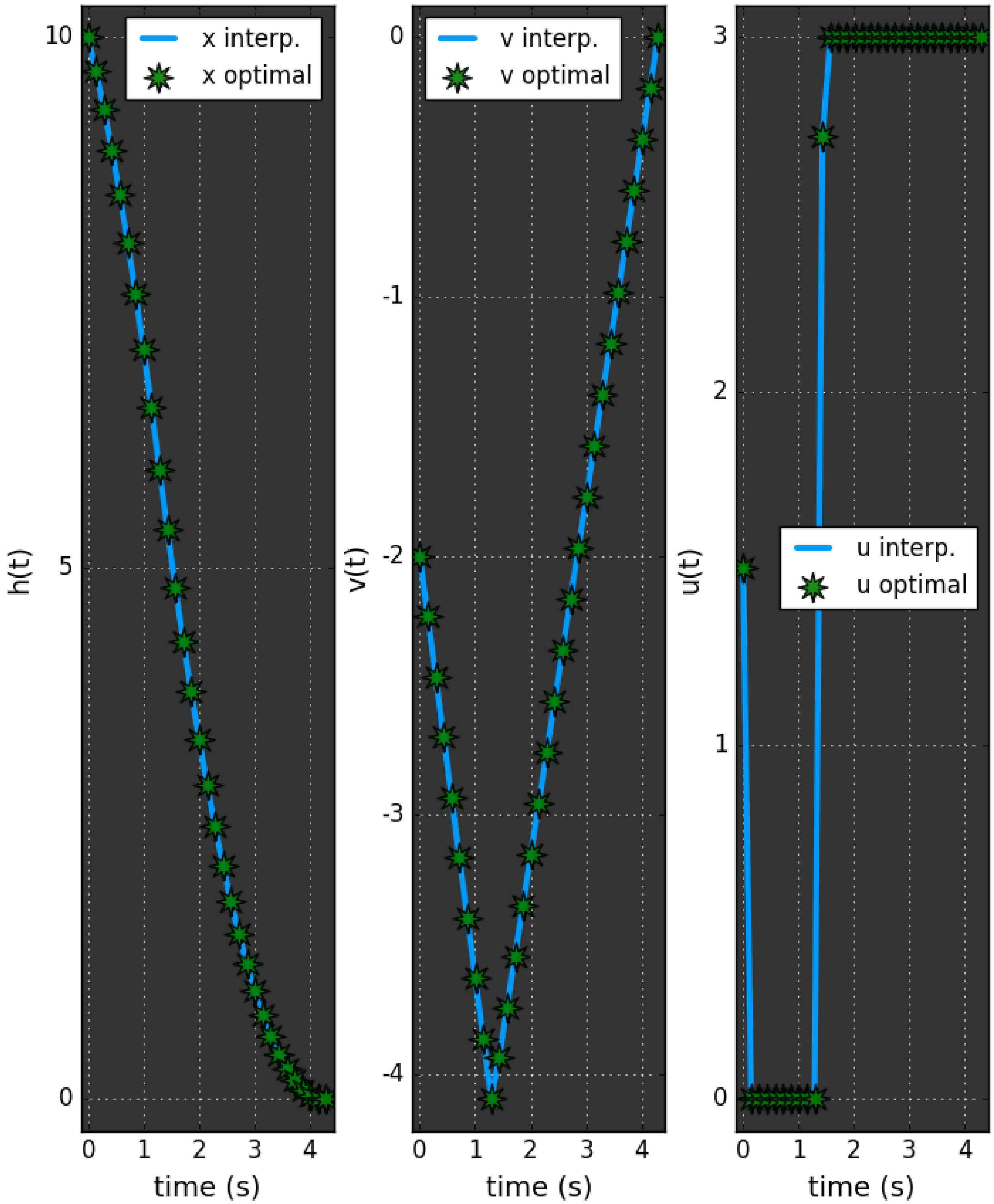
`N = 30` –> ex#1

LGR



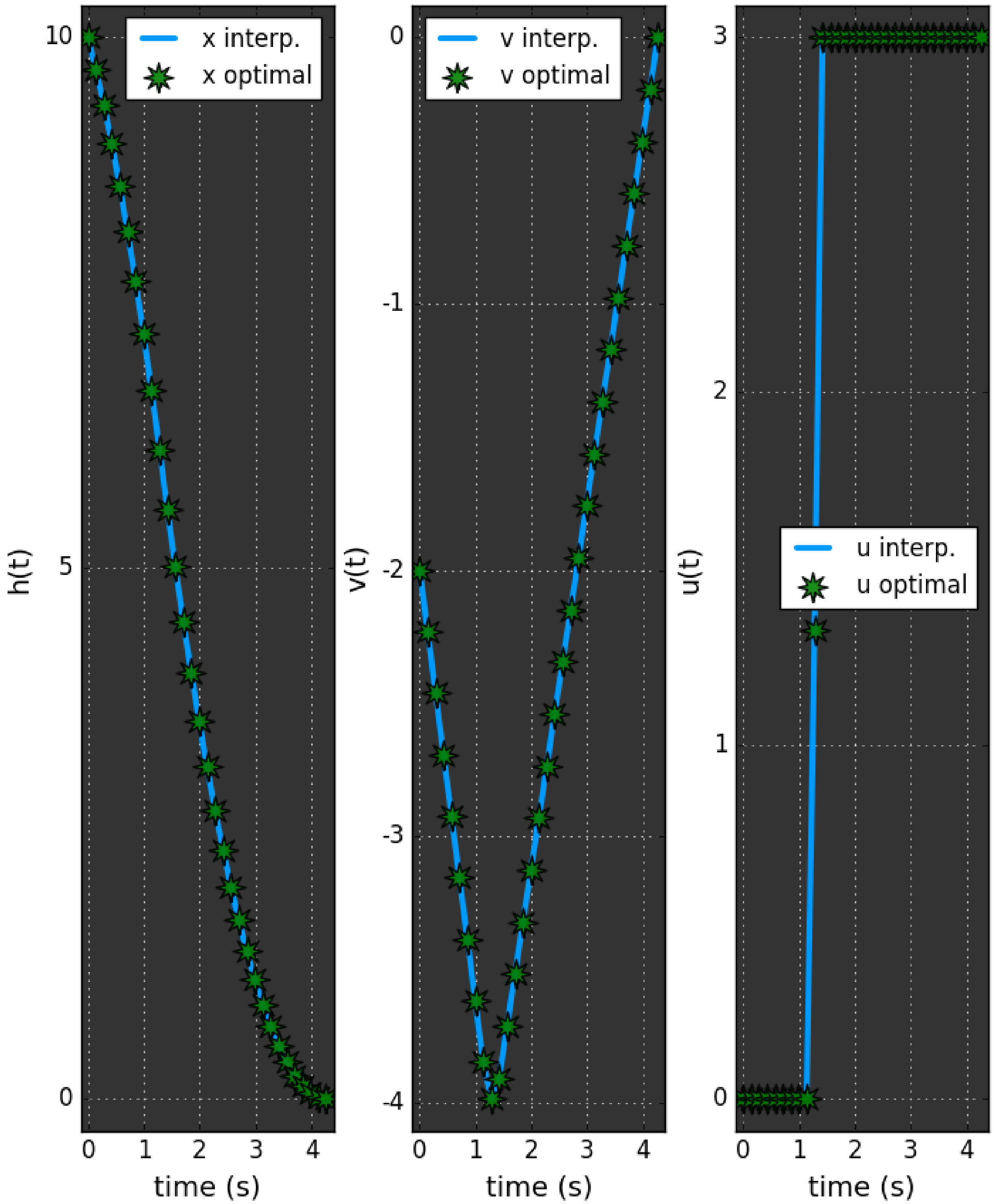
x interp.
x optimal
h(t)
time (s)
v interp.
v optimal
v(t)
time (s)
u interp.
u optimal
u(t)
time (s)

Bkw Euler



x interp.
x optimal
h(t)
time (s)
v interp.
v optimal
v(t)
time (s)
u interp.
u optimal
u(t)
time (s)

Trapezoidal



Package Functionality

Optimal Control Problem Definition

Bibliography

Bibliography

[BGar11] Divya Garg. *Advances in global pseudospectral methods for optimal control*. PhD thesis, University of Florida, 2011.