

---

# NLOptControl Documentation

*Release 0.0.1-rc1*

**Huckleberry Febbo**

December 25, 2016



<b>1</b>	<b>Table of Contents</b>	<b>1</b>
1.1	Background Information . . . . .	1
1.2	Package Functionality . . . . .	6
1.3	Bibliography . . . . .	16
	<b>Bibliography</b>	<b>17</b>



---

## Table of Contents

---

### 1.1 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.

#### 1.1.1 Direct Transcription of Optimal Control Problems

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

#### 1.1.2 Time Marching Methods

**Euler Method**

**Trapezoidal Method**

#### 1.1.3 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  $\tau$ , 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:

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

- 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:

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

- So, the function evaluated at  $\tau_k$  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:

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

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

- values of  $\tau$
- 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:

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

This can be approximated using quadrature:

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

where  $w_k$  are quadrature weights and depend only on:

- values of  $\tau$
- type of interpolating polynomial

## Legendre Pseudospectral Method

- Polynomial

Define an N order Legendre polynomial as:

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

- Nodes

$$\tau_k = \begin{cases} -1, & \text{if } k = 0 \\ \text{kth root of } \dot{L}_{N_t}(\tau), & \text{if } k = 1, 2, 3, \dots, N_t - 1 \\ +1 & \text{if } k = N_t \end{cases} \quad (1.1)$$

- Differentiation Matrix
- Interpolating Polynomial Function

### 1.1.4 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:

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

- Inequality Path Constraints:

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

- Integral Constraints:

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

- Event Constraints:

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

### 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  $\tau$ , 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  $\tau$  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:

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

- Inequality Path Constraints:

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

- Integral Constraints:

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

- Event Constraints:

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

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

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

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

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

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



## 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:

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

- Inequality Path Constraints:

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

- Integral Constraints:

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

- Event Constraints:

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

- State Continuity

- Also, we must **now** constrain the state to be continuous at each interior mesh point  $(T_1, \dots, T_{K-1})$  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. 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  $k \in [1, \dots, K]$ :

$$\mathbf{x}^{(k)}(\tau) \approx \mathbf{X}^{(k)}(\tau) = \sum_{j=1}^{N_k+1} \mathbf{X}_j^{(k)} \frac{d\ell_j^k(\tau)}{d\tau} \quad (1.2)$$

$$\text{where,} \quad (1.3)$$

$$\ell_j^k(\tau) = \prod_{\substack{l=1 \\ l \neq j}}^{N_k+1} \frac{\tau - \tau_l^{(k)}}{\tau_j^{(k)} - \tau_l^{(k)}} \quad (1.4)$$

also,

- $\ell_j^{(k)}(\tau)$ , ( $j = 1, \dots, N_k + 1$ ) is a basis of Lagrange polynomials
- $(\tau_1^{(k)}, \dots, \tau_{N_k}^{(k)})$  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

## 1.2 Package Functionality

### 1.2.1 Code Development

#### Approximation of Optimal Control Problem

##### Completed Functionality

##### Legendre Gaussian Method

##### Single Interval

**Basic Problem Definition** The code developed in this package uses the Legendre-Pseudospectral Method with Lagrange-Gauss-Lobatto (LGL) nodes. A basic description of this implementation presented in this documentation at [Pseudospectral Methods](#) and more much more detailed information can be found in both [\[A2\]](#) and [\[A1\]](#).

##### Examples

In these examples we use:

- Legendre-Gauss-Lobatto (LGL) nodes
- Single interval approximations
- Approximate integrals in the range of  $[-1, 1]$
- Approximate derivatives in the range of  $[-1, 1]$

These examples can be:

- Viewed remotely on using the [jupyter nbviewer](#).
- Viewed locally and interacted using IJulia

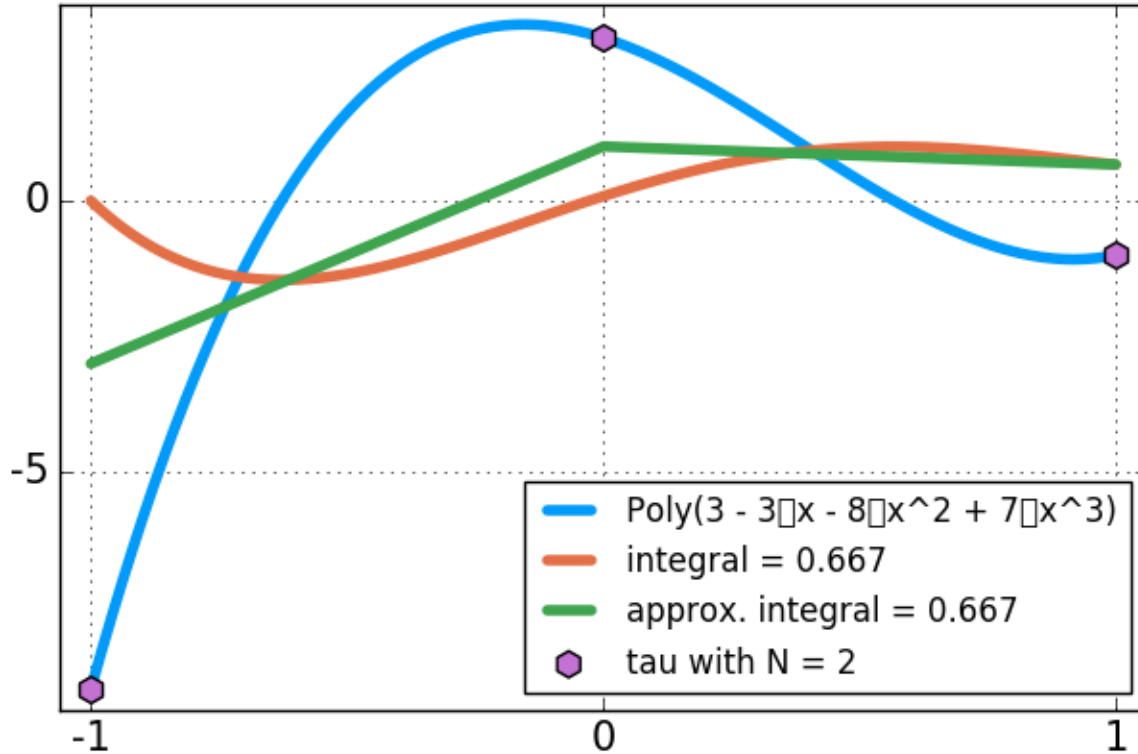
To do this in julia type:

```
using IJulia
notebook(dir=Pkg.dir("NLOptControl/examples/LGL_SI/"))
```

**Example 1** In the first example, we borrow a problem from [Wikipedia](#).

where:

$$y(x) = 7x^3 - 8x^2 - 3x + 3$$



#### Difference between the Wikipedia Example and this Example

The difference between Wikipedia example and this one is that the Wikipedia example uses Gauss-Legendre Quadrature while the code developed in this package uses Legendre-Pseudospectral Method with Lagrange-Gauss-Lobatto (LGL) nodes. Information on the difference between these methods and many more can be found in both [\[B2\]](#) and [\[B1\]](#).

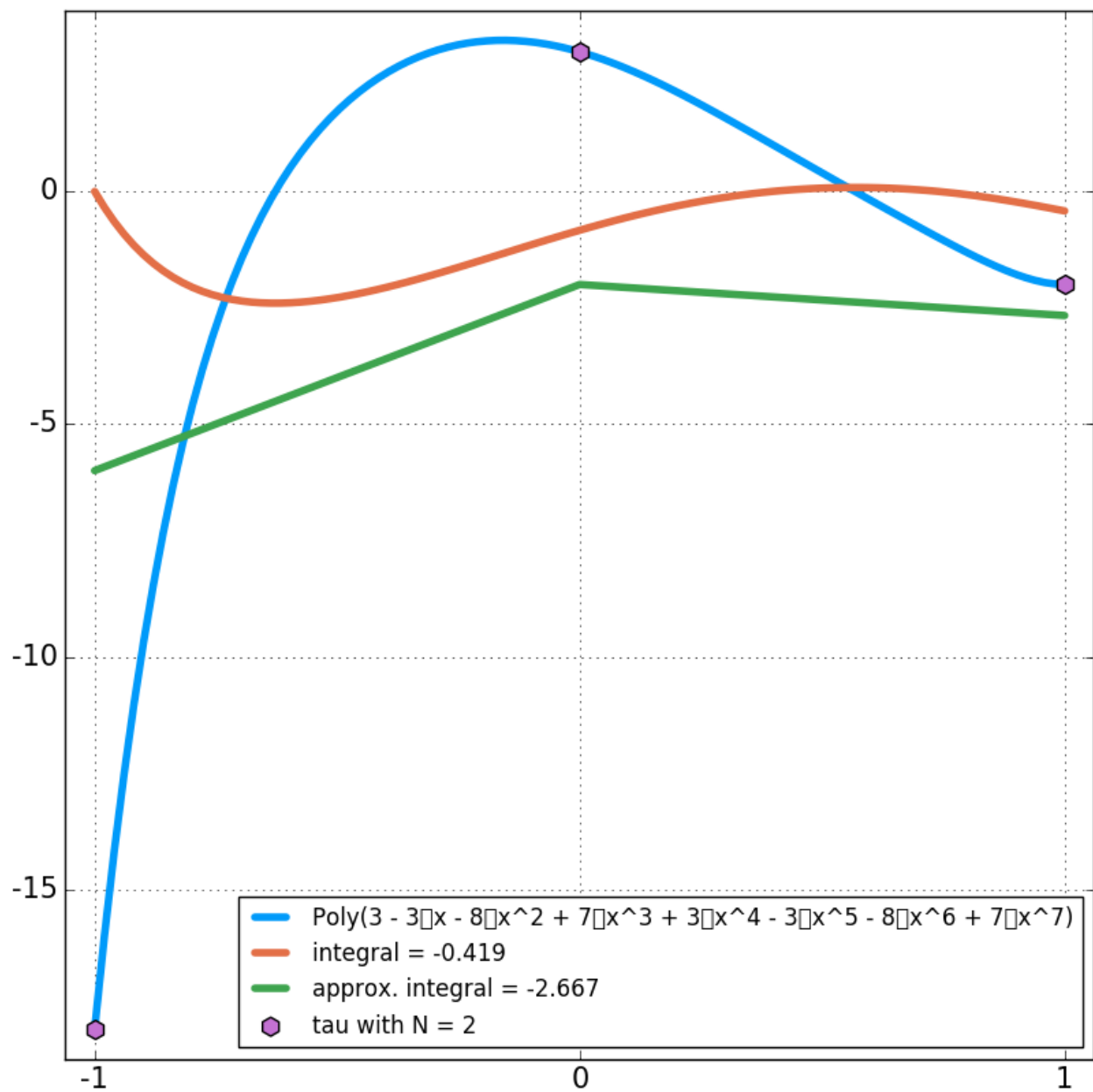
- Conclusions
  - We are able to exactly determine the integral

#### References

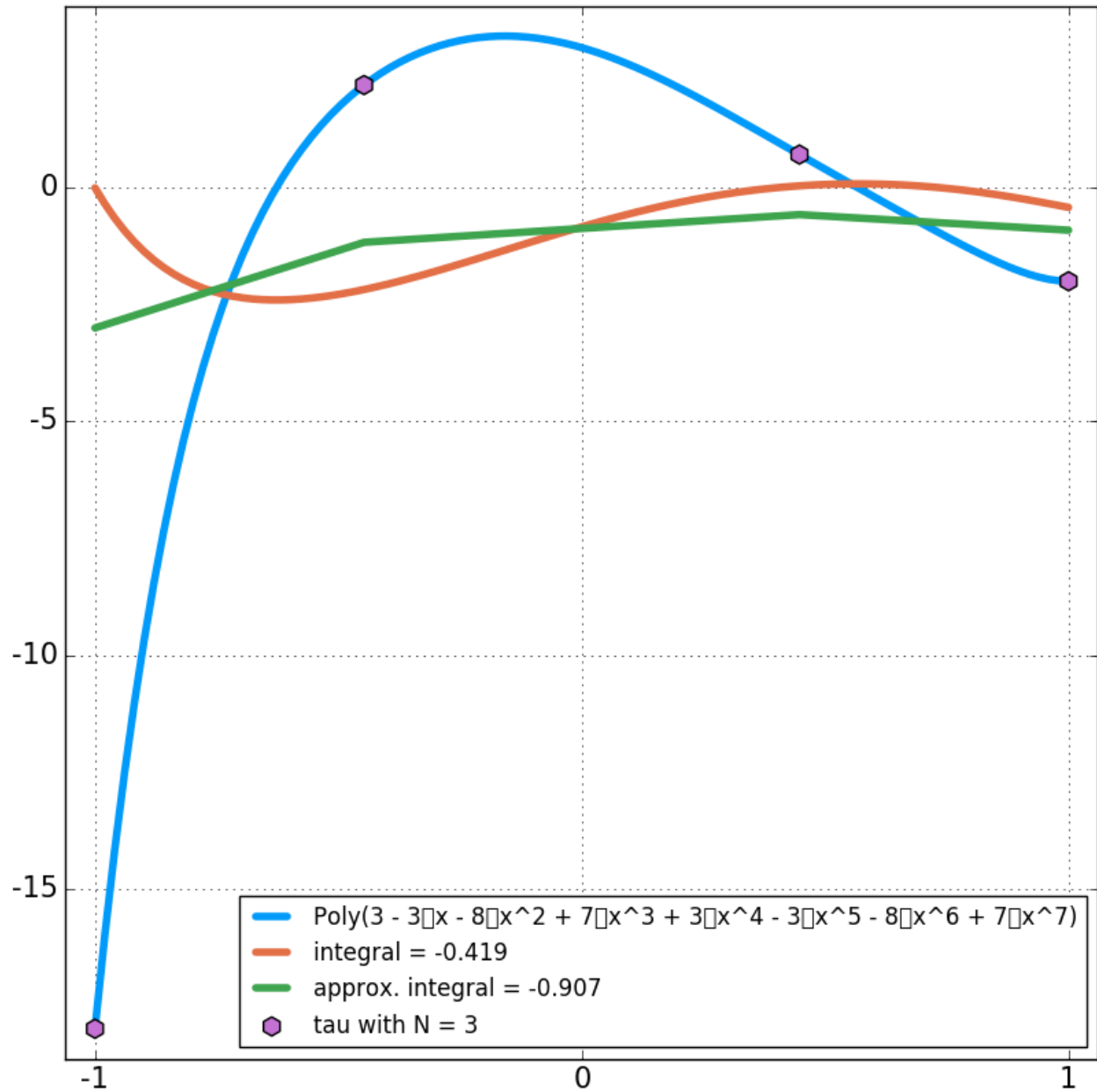
**Example 2** In the second example, we increase the order of the polynomial from 3 to 7. Then we increase  $N$  until we achieve accurate enough results.

where:

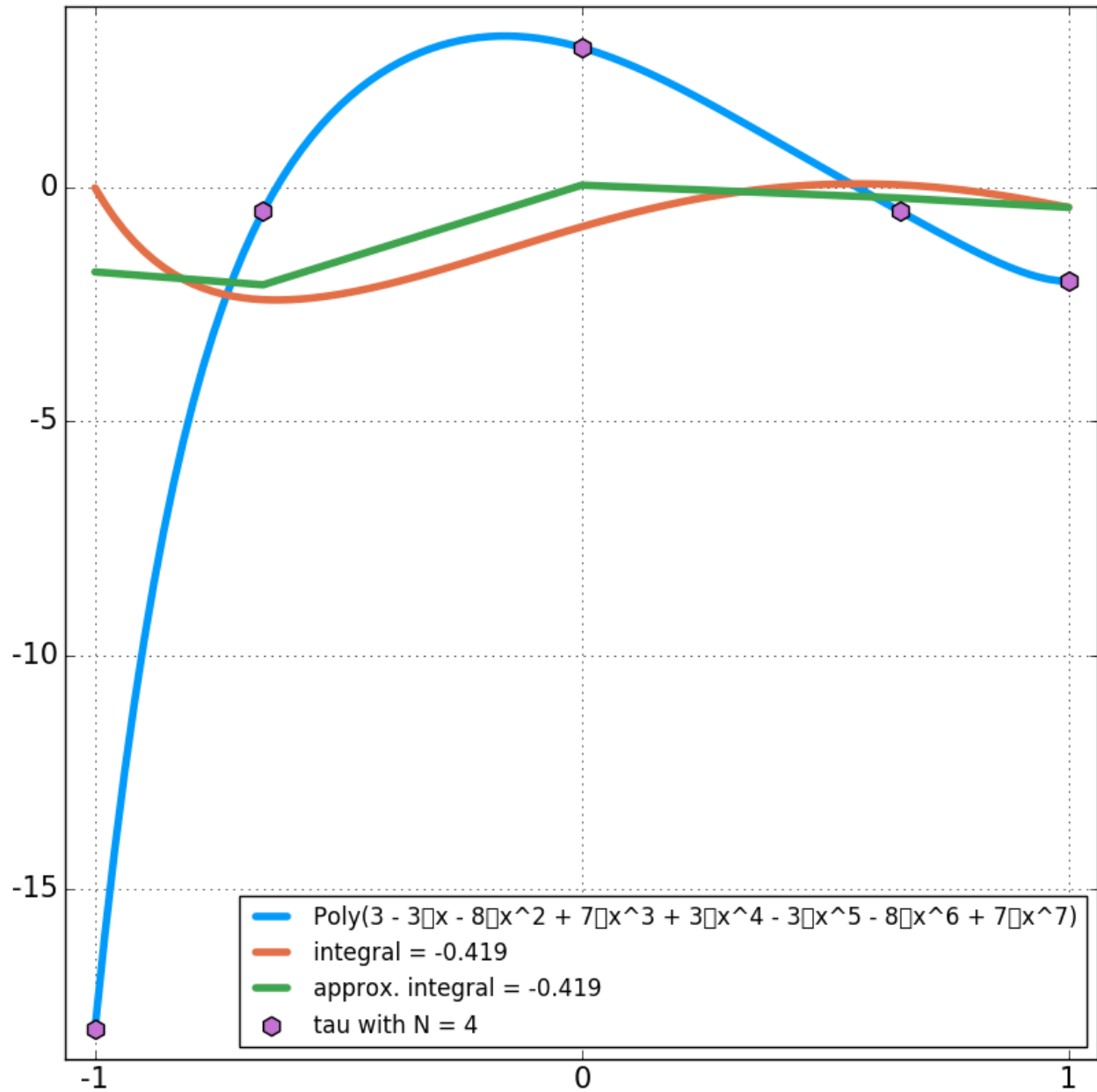
$$y(x) = 7x^7 - 8x^6 - 3x^5 + 3x^4 + 7x^3 - 8x^2 - 3x + 3$$



Test 2a



Test 2b



Test 2c

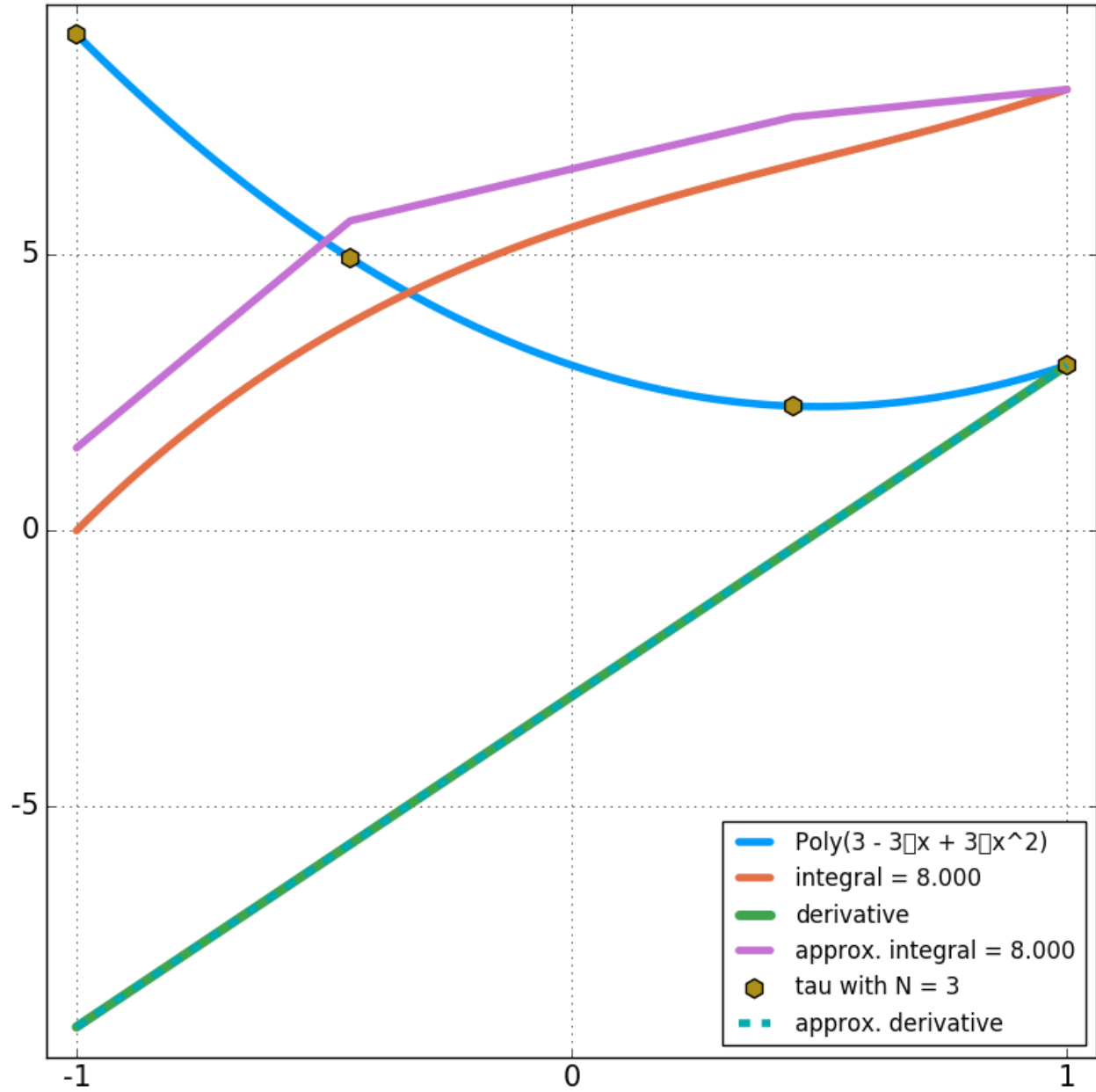
### Conclusions

- We are able to determine the integral very accurately when we increase  $N$  to 4

**Example 3** In the third example, we approximate the derivative of a function by *Approximating Derivatives*.

**Test 3a** In this test:

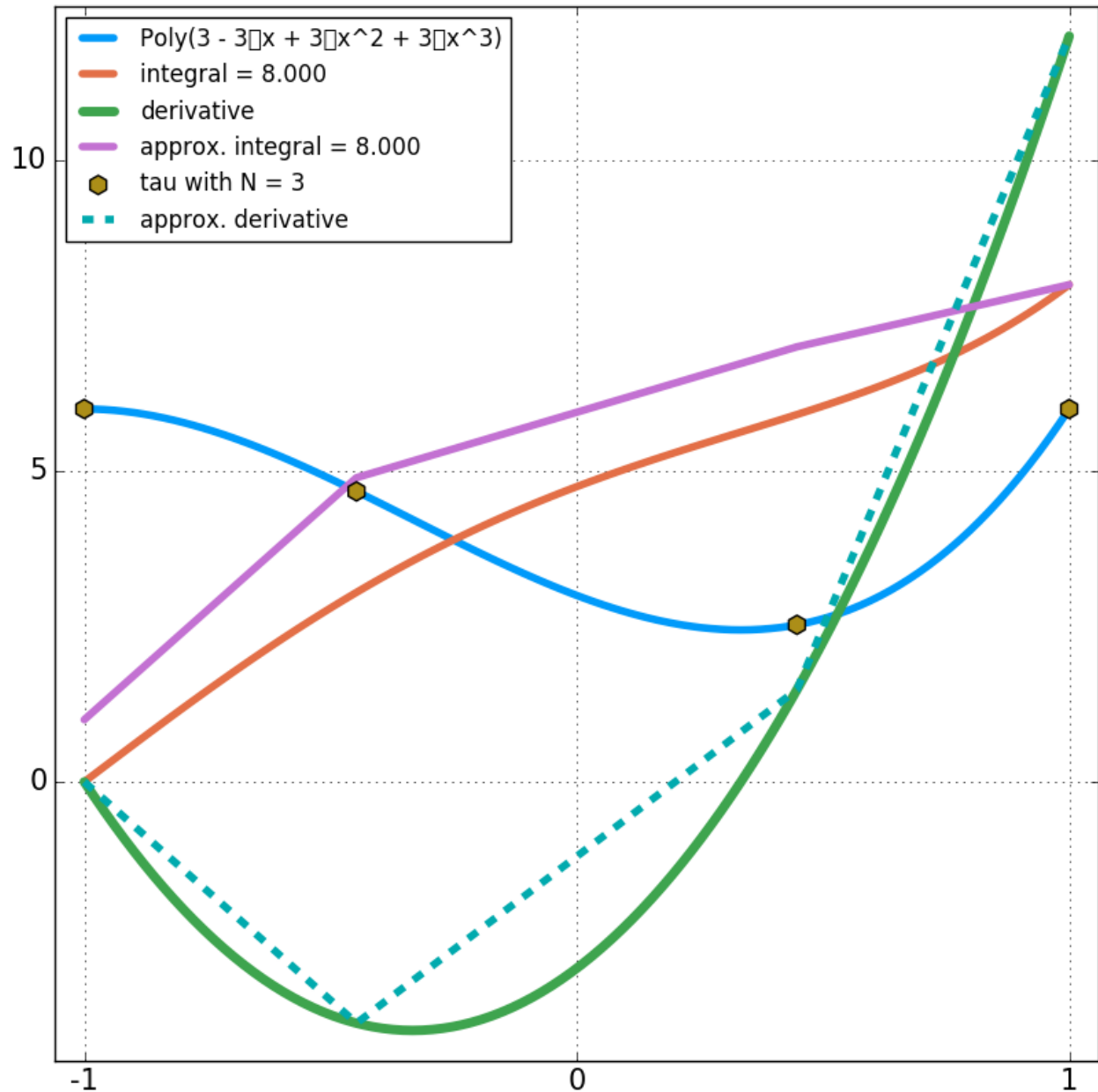
$$y(x) = 3x^2 - 3x + 3$$



- We are able to determine the derivative exactly when they are linear functions is  $N = 3$

**Test 3b** In this test we increase the order of  $y(x)$  to:

$$y(x) = 3x^3 + 3x^2 - 3x + 3$$

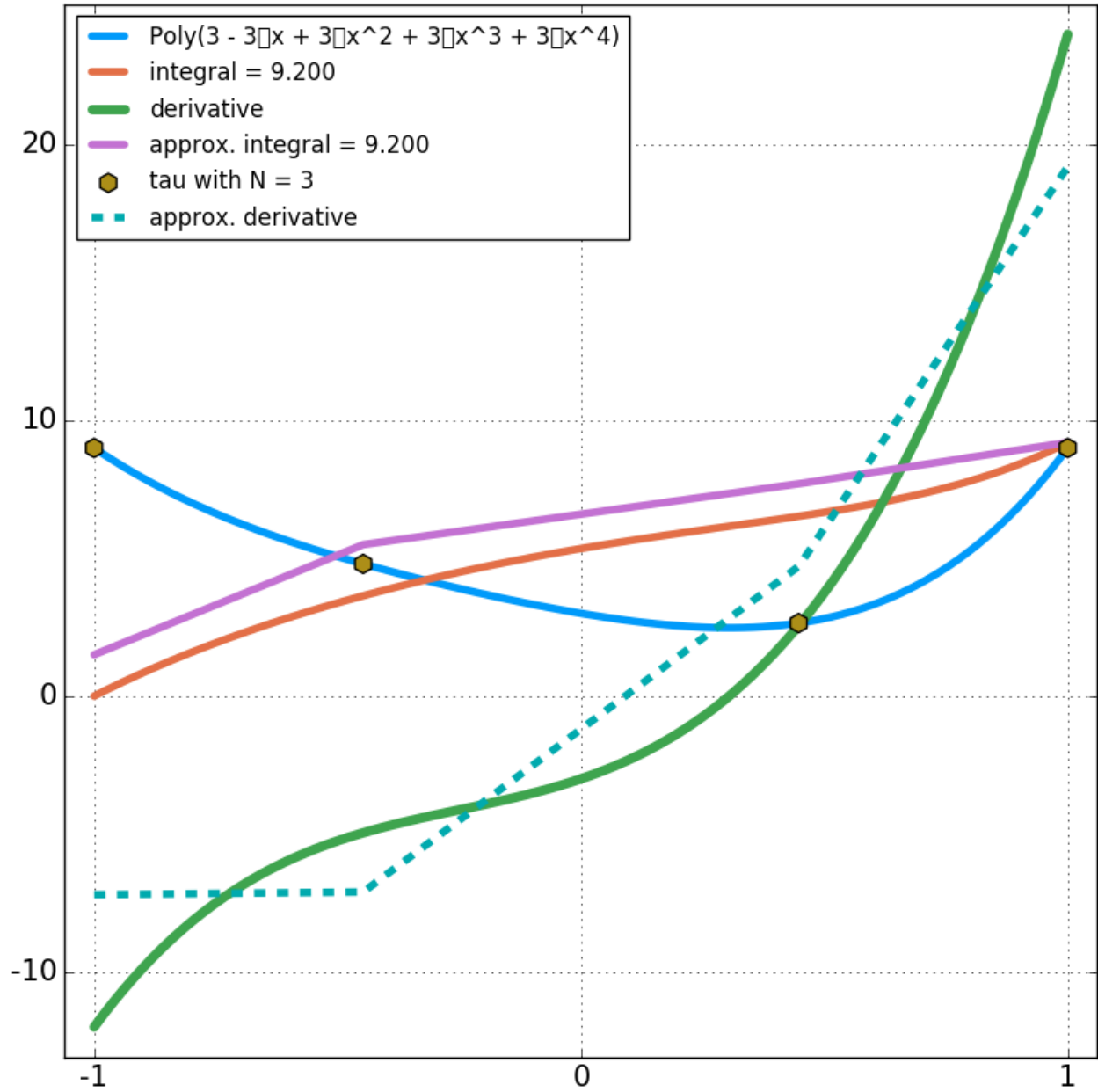


- When the derivative function becomes nonlinear
  - We can no longer calculate it exactly everywhere
  - There are only  $N_{t+1} = 4$  node points
  - To calculate the derivative exactly we would need an amount of  $N_{t+1}$
- We are still calculating the integral exactly and should be able to with  $N = 3$  until  $x^5$

**Test 3c** In this test we increase the order of  $y(x)$  to:

$$y(x) = 3x^4 + 3x^3 + 3x^2 - 3x + 3$$

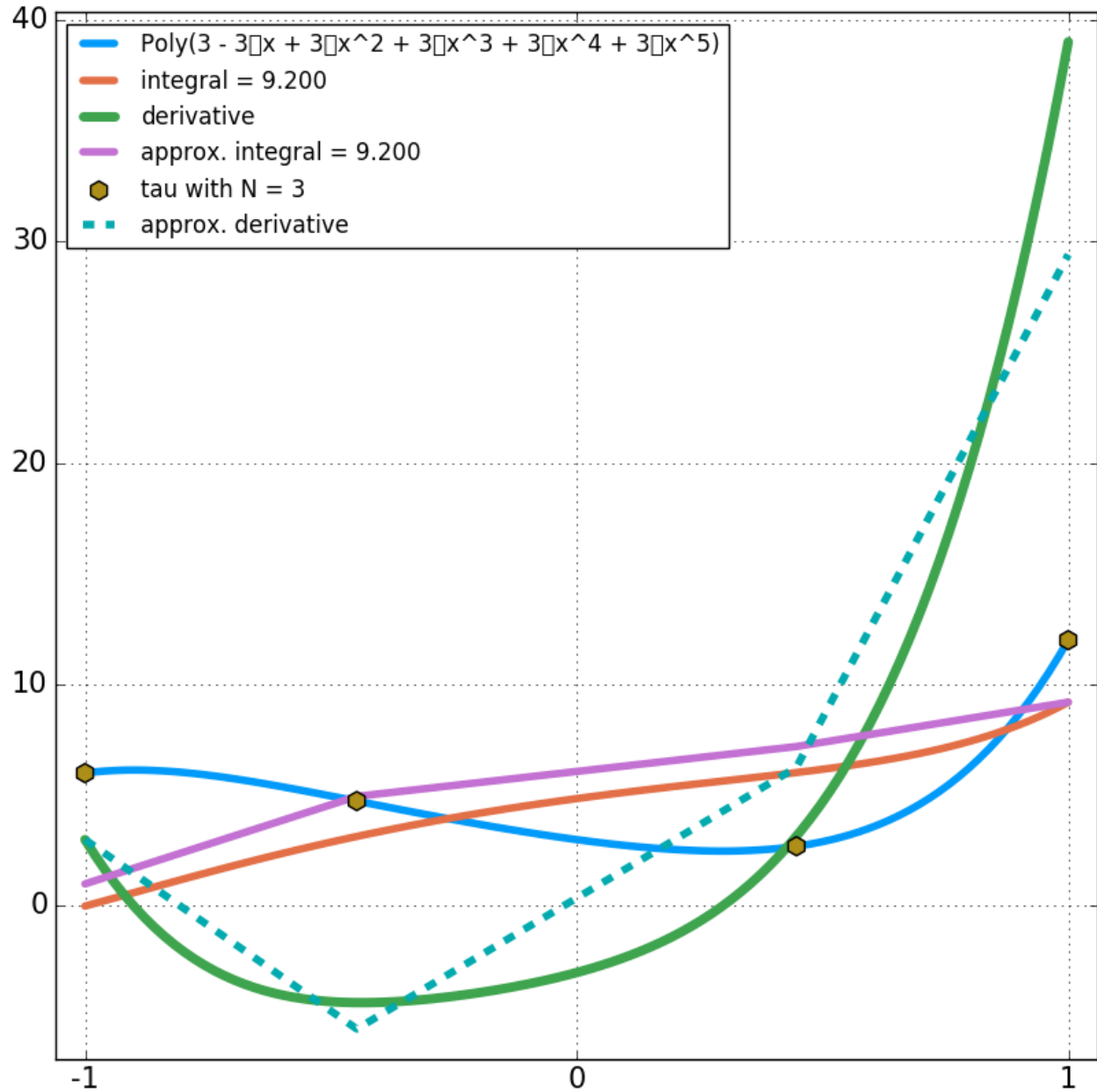




- We are still calculating the integral exactly and should be able to with  $N = 3$  until  $x^5$

**Test 3d** In this test we increase the order of  $y(x)$  to:

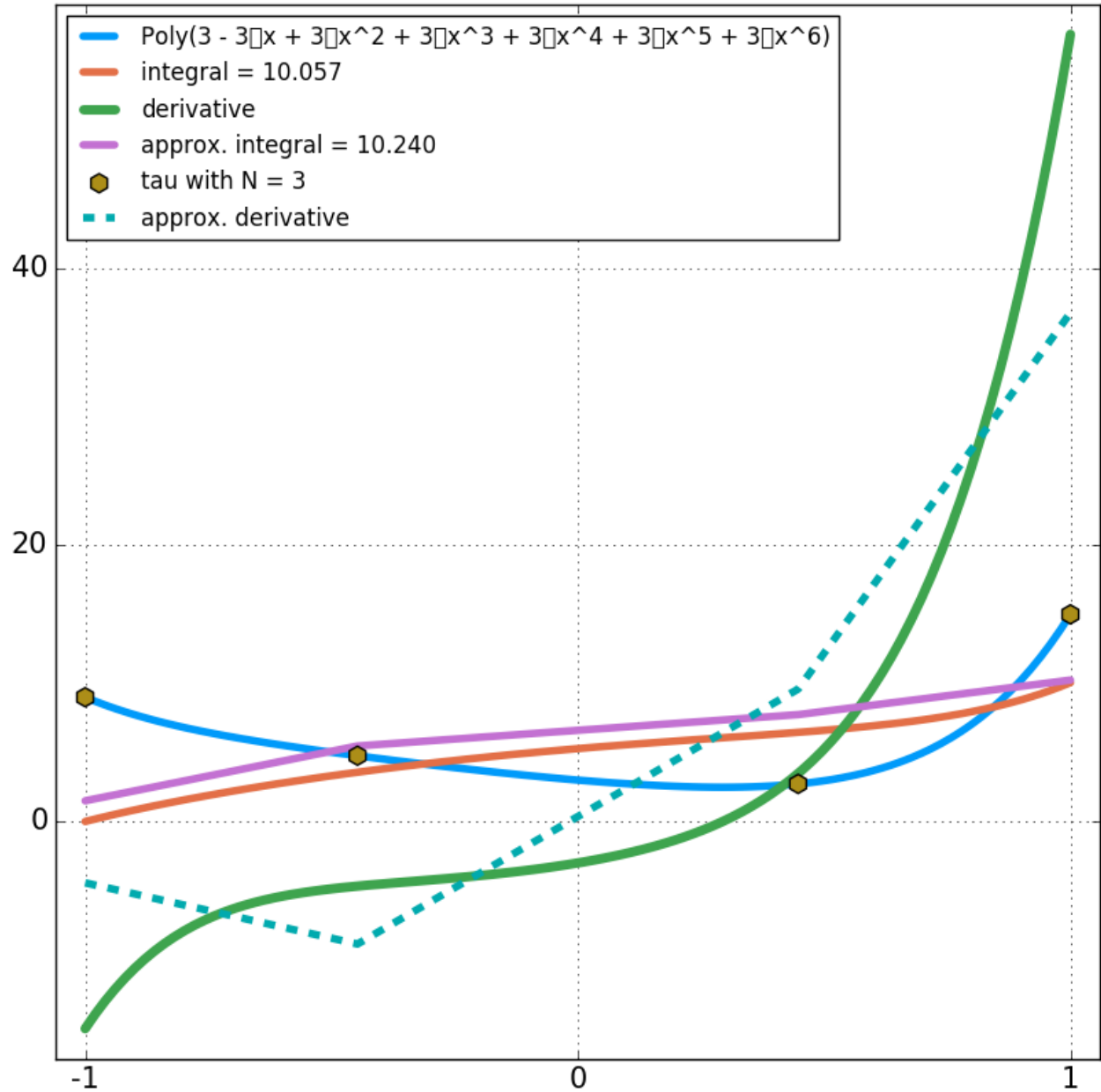
$$y(x) = 3x^5 + 3x^4 + 3x^3 + 3x^2 - 3x + 3$$



- We are still calculating the integral exactly with  $N = 3!!$
- The percent error is = 0.000000000000000000 %

**Test 3e** In this test we increase the order of  $y(x)$  to:

$$y(x) = 3x^6 + 3x^5 + 3x^4 + 3x^3 + 3x^2 - 3x + 3$$



- As expected, we are not still calculating the integral exactly with  $N = 3$ !!
- The percent error is  $= -1.818181818181822340 \%$

## References

## Developing Code

This site documents current progress and functionality that is being developed.

## Current Focus

**Old Records** This is for documentation that was created where something was still being worked on where:

1. Everything was not working as expected, but has now been fixed or is obsolete.
2. Removed Functionality

## 1.3 Bibliography

- [B1] Daniel Ronald Herber. Basic implementation of multiple-interval pseudospectral methods to solve optimal control problems. *UIUC-ESDL-2015-01*, 2015.
- [B2] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*. volume 41. Springer Science & Business Media, 2011.
- [A1] Daniel Ronald Herber. Basic implementation of multiple-interval pseudospectral methods to solve optimal control problems. *UIUC-ESDL-2015-01*, 2015.
- [A2] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*. volume 41. Springer Science & Business Media, 2011.