# Dynamic optimization

From optimization

Authors: Hanyu Shi (ChE 345 Spring 2014)

Steward: Dajun Yue, Fengqi You

Date Presented: Apr. 10, 2014

Authors: Issac Newton, Albert Einstein (ChE 345 Spring 2014)

Steward: Dajun Yue, Fengqi You

Date Presented: Apr. 10, 2014

## Contents

## Introduction

In this work, we will focus on the “at the same time” or direct transcription approach which allow a simultaneous method for the dynamic optimization problem. In particular, we formulate the dynamic optimization model with orthogonal collocation methods. These methods can also be regarded as a special class of implicit Runge–Kutta (IRK) methods. We apply the concepts and properties of IRK methods to the differential equations directly. With locating potential break points appropriately, this approach can model large-scale optimization formulations with the property of maintaining accurate state and control profiles. We mainly follows Biegler's work.

## General Dynamic Optimization Problem

Differential algebraic equations in process engineering often have following characteristics: first,large-scale models – not easily scaled; second, sparse but no regular structure; third, direct linear solvers widely used; last, coarse-grained decomposition of linear algebra.

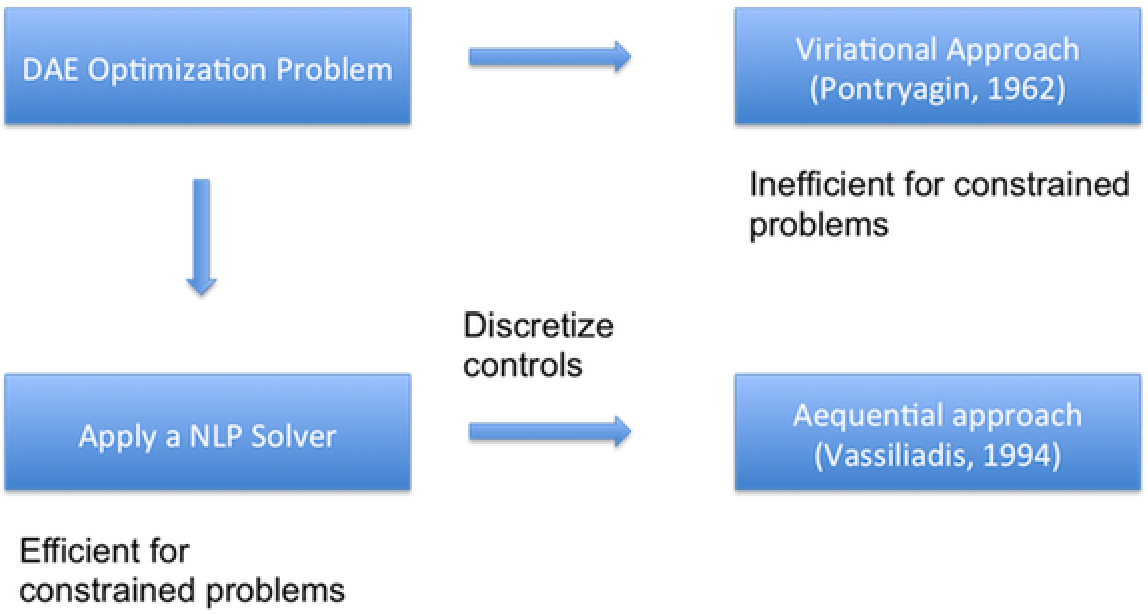

DAE Optimization Problem
Viriational Approach (Pontryagin, 1962)
Inefficient for constrained problems
Discretize controls
Apply a NLP Solver
Aequential approach (Vassiliadis, 1994)
Efficient for constrained problems

Figure 2. Dynamic optimization approach

There are several approaches can be applied to solve the dynamic optimization problems, which are shown in Figure 2.

Differential equations can usually be used to express conservation Laws, such as mass, energy, momentum. Algebraic equations can usually be used to express constitutive equations, equilibrium, such as physical properties, hydraulics, rate laws. Algebraic equations usually have semi-explicit form and assume to be index one i.e., algebraic variables can be solved uniquely by algebraic equations.

Dynamic Optimization Problem has the following general form:

$$\min\ \Phi\left(z\left(t\right),y\left(t\right),u\left(t\right),p,t_f\right)$$
$$s.t.\ \ \frac{dz(t)}{dt}=f\left(z\left(t\right),y\left(t\right),u\left(t\right),p\right)$$
$$g\left(z\left(t\right),y\left(t\right),u\left(t\right),p\right)=0$$
$$z^0=z\left(0\right)$$
$$z^l\leq z\left(t\right)\leq z^u$$
$$y^l\leq y\left(t\right)\leq y^u$$
$$u^l\leq u\left(t\right)\leq u^u$$
$$p^l\leq p\leq p^u$$

$t$, time

$z$ , differential variables y, algebraic variables

$t_f$ , final time

$u$ , control variables

$p$ , time independent parameters

(This follows Biegler's slides ）

## Derivation of Collocation Methods

We first consider the differential algebraic system shown as follows:

$$\frac{dz}{dt} = f\left(z\left(t\right), y\left(t\right), u\left(t\right), p\right),\ z\left(0\right) = z_0 \quad (1)$$
$$g\left(z\left(t\right), y\left(t\right), u\left(t\right), p\right) = 0$$

The simultaneous approach requires discretizing of the state variables $z\left(t\right)$, output variables $y\left(t\right)$ and manipulate variables $u\left(t\right)$. We require the following properties to yield an efficient NLP formulation:

1) The explicit ODE discretization holds little computational advantage because Since the nonlinear program requires an iterative solution of the KKT conditions.

2) A single step approach which is self-starting and does not rely on smooth profiles that extend over previous time steps is preferred, because the NLP formulation needs to deal with discontinuities in control profiles.

3) The high-order implicit discretization provides accurate profiles with relatively few finite elements. As a result, the number of finite elements need not be excessively large, particularly for problems with many states and controls.

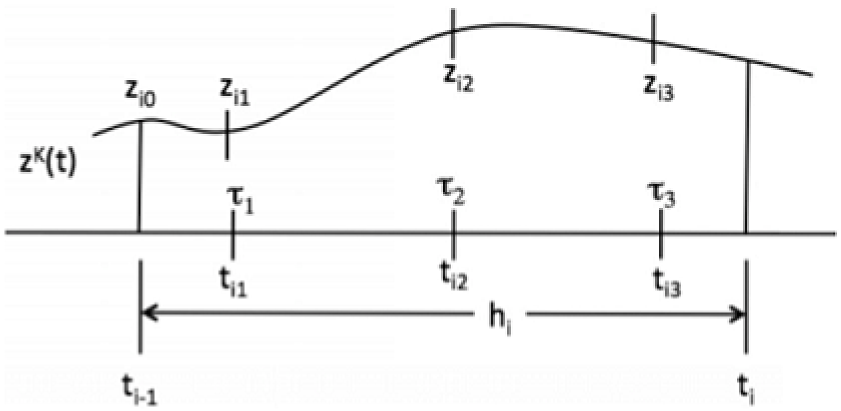


Figure 1: Polynomial approximation for state profile across a finite element.

## Polynomial Representation for ODE Solutions

We consider the following ODE:

$$\frac{dz}{dt} = f\left(z\left(t\right), t\right),\ z\left(0\right) = z_0 \quad (2)$$

to apply the collocation method, we need to solve the differential equation (2) at certain points. For the state variable, we consider a polynomial approximation of order $K+1$ (i.e., degree ≤ $K$ ) over a single finite element, as shown in the above figure. This polynomial, denoted by $z^K(t)$, can be represented in a number of equivalent ways, including the power series representation shown in equation (3), the Newton divided difference approximation, or B-splines.

$$\frac{dz}{dt} = f\left(z\left(t\right),t\right),\ z\left(0\right) = z_0 \quad (3)$$

We apply representations based on Lagrange interpolation polynomials to generate the NLP formulation, because the polynomial coefficients and the profiles have the same variable bounds. Here we select $K+1$ interpolation points in element i and represent the state in a given element $i$ as

$$\left.\begin{aligned} t &= t_{i-1} + h_i \cdot \tau, \\ z^K\left(t\right) &= \sum_{j=0}^{K} l_j\left(\tau\right) \cdot z_{ij}, \end{aligned}\right\} t \in \left[t_{i-1}, t_i\right], \tau \in \left[0,1\right], \quad (4)$$
$$where\ l_j\left(\tau\right) = \prod_{k=0,\neq j}^{K} \frac{\left(\tau-\tau_k\right)}{\left(\tau_j-\tau_k\right)},$$

$\tau_0 = 0,\ \tau_i < \tau_{i+1},\ j = 0, ..., K-1$, and hi is the length of element $i$ . This polynomial representation has the desirable property that $z^K(t_{ij}) = z_{ij}$, where $t_{ij} = t_{i-1} + \tau_j h_j$.

We use a Lagrange polynomial with K interpolation points to represent the time derivative of the state. This leads to the Runge–Kutta basis representation for the differential state:

$$z^K\left(t\right) = z_{i-1} + h_i \cdot \sum_{j=1}^{K} \Omega_j\left(\tau\right) \cdot \dot{z}_{ij} \quad (5)$$

where $z_{i-1}$ is a coefficient that represents the differential state at the beginning of element $i$ , $\dot{z}_{ij}$ represents the time derivative $\frac{dz^K(t_{ij})}{d\tau}$, and $\Omega_j(\tau)$ is a polynomial of order K satisfying

$$\Omega_j\left(\tau\right) = \int_0^\tau l_j(\tau')d\tau', t \in \left[t_{i-1}, t_i\right], \tau \in \left[0,1\right] \quad (6)$$

We substitute the polynomial into equation (1) to calculate the polynomial coefficients, which is an approximation of the DAE. This results in the following collocation equations.

$$z^K(t) = z_{i-1} + h_i \cdot \sum_{j=1}^{K} \Omega_j(\tau) \cdot \dot{z}_{ij} \quad (7)$$

with $z^k(t_i - 1)$ calculated separately. For the polynomial representations (4) and (5), we normalize time over the element, write the state profile as a function of τ , and apply $\frac{dz^K}{d\tau} = h_i \frac{dz^K}{dt}$ easily. For the Lagrange polynomial (4), the collocation equations become

$$\sum_{j=0}^{K} z_{ij} \cdot \frac{dl_j(\tau_k)}{d\tau} = h_i \cdot f(z_{ik}, t_{ik}),\ k = 1, ..., K \quad (8)$$

while the collocation equations for the Runge–Kutta basis are given by

$$\dot{z}_{ik} = f(z_{ik}, t_{ik}) \quad (9)$$

$$z_{ik} = z_{i-1} + h_i \cdot \sum_{j=1}^{K} \Omega_j(\tau) \cdot \dot{z}_{ij},\ k = 1, ..., K \quad (10)$$

with $z_i - 1$ determined from the previous element $i - 1$ or from the initial condition on the ODE.

## Example

An example is given here to demonstrate the application of the collocation method.

A differential equation is given as follows:

$$\frac{dz}{dt} = z^2 - 2 \cdot z + 1,\ z(0) = -3 \quad (11)$$

With t \in \left[ {0,1} \right], The analytic solution of this differential equation is $z(t) = \frac{4 \cdot t - 3}{4 \cdot t + 1}$.

Lagrange interpolation and collocation method is applied to this differential equation respectively. And the number of collocation points in each finite element is 3. The number of finite elements is N , and the length of the finite element is 1/N . The following equations is given then:

$$\sum_{j=0}^{3} z_{ij} \frac{dl_j(\tau_k)}{d\tau} = h\left(z_{ik}^2 - 2 \cdot z_{ik} + 1\right),\ k = 1, ..., 3,\ i = 1, ..., N \quad (12)$$

$$z_{i+1,0} = \sum_{j=0}^{0} l_j\left(1\right) \cdot z_{ij},\ i = 1, ..., N-1 \quad (13)$$

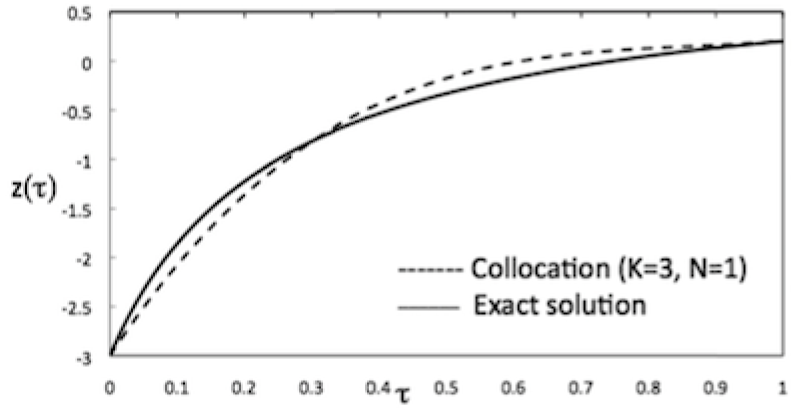
$$z_f = \sum_{j=0}^{K} l_j\left(1\right) \cdot z_{Nj},\ z_{1,0} = -3 \quad (14)$$

With Radau collocation method, $\tau_0 = 0$, $\tau_1 = 0.155051$, $\tau_2 = 0.644949$ and $\tau_3 = 1$ can be obtained. The collocation equations are given as follows:

$$\sum_{j=0}^{3} z_j \frac{dl_j\left(\tau_k\right)}{d\tau} = \left(z_k^2 - 2 \cdot z_k + 1\right),\ k = 1, ..., 3 \quad (14)$$

which can be formulated as:

$$\begin{aligned} &z_0 \cdot \left(-30 \cdot \tau_k^2 + 36 \cdot \tau_k - 9\right) + z_1 \cdot \left(46.7423 \cdot \tau_k^2 - 51.2592 \cdot \tau_k + 10.0488\right) \\ &+z_3 \cdot \left(-26.7423 \cdot \tau_k^2 + 20.5925 \cdot \tau_k - 1.38214\right) + z_3 \cdot \left(10 \cdot \tau_k^2 - \tfrac{16}{3} \cdot \tau_k + \tfrac{1}{3}\right) \quad (15) \\ &= \left(z_k^2 - 2 \cdot z_k + 1\right),\ k = 1, ..., 3 \end{aligned}$$



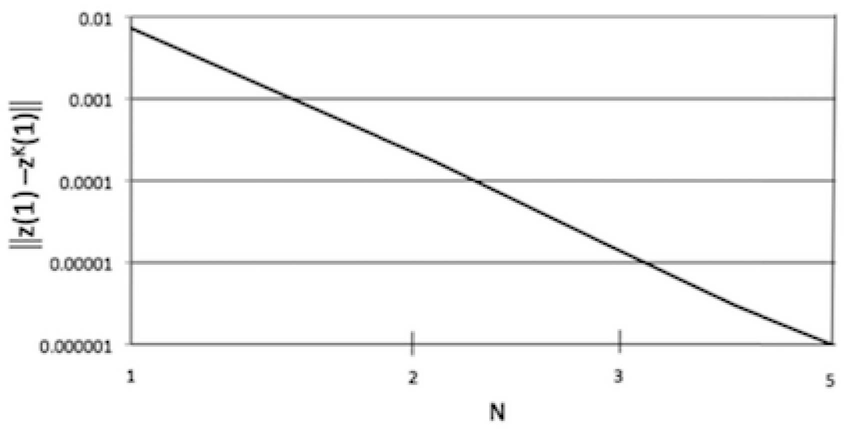


Figure 3. Comparison of Radau collocation solution with exact solution

The results are given as following by solving the above equations:

$$\begin{cases} z_0 = -3 \\ z_1 = -1.65701 \\ z_2 = 0.032053 \\ z_3 = 0.207272 \end{cases} \quad (16)$$

As shown in Figure 3.2 the error $\left\| z(1) - z^K(1) \right\|$, is less than $10^-6$ for $N = 5$ and converges with $O(h^5)$, which is consistent with the expected order $2K - 1$.

(This example follows the work of Biegler and can be found in P293 of " Nonlinear Programmng".)

## Conclusion

In this work, we mainly discussed simultaneous collocation approach for dynamic optimization problems, which formulated the differential equations to a set of algebraic equations. These direct transcription formulations depended on fully discretizing of the differential algebraic equations (DAE), which enabled us solve the simultaneous optimization problem without relying on embedded DAE solvers. Because of this simultaneous formulation, we got the exact first and second order derivatives through the optimization modeling system, and both structure and sparsity can be exploited.

## References

1. Biegler, Lorenz T. Nonlinear programming: concepts, algorithms, and applications to chemical processes. Vol. 10. SIAM, 2010.

2. Chu, Yunfei, and Fengqi You. "Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm." Computers & Chemical Engineering 47 (2012): 248-268.

3. http://en.wikipedia.org/wiki/Dynamic_programming

4. http://en.wikipedia.org/wiki/Differential_algebraic_equation

5. http://numero.cheme.cmu.edu/uploads/dynopt.pdf