

Review of “Survey of Numerical Methods for Trajectory Optimization” (Betts 1998)

In this survey, Betts describes indirect methods that rely on finding the NOCs, and then solving a non-linear multipoint boundary value problem, as well as direct methods which do not require an analytic representation of the NOCs or initial guesses for the adjoint variables, but instead directly optimize the objective function using some parametric representation for the control variables. He stresses that both methods are similar in the sense that they use a Newton-based iteration to adjust a finite set of variables. Indirect methods have three main drawbacks – they need to solve complicated expressions for the NOCs, the region of convergence can be small, and the initial guess must be very good. Direct methods directly optimize the objective without the need for explicit NOCs and methods like direct shooting and direct transcription are widely used.

Broadly, talking about indirect and direct methods, Betts makes several interesting points. Firstly, in an indirect method, we often choose the absolute value of the gradient as a merit function, i.e. if the gradient absolute value decreases, we assume we’re moving towards the local optimum. However, for a direct method, we need to be more careful and we take a step such that the objective function itself decreases. This means that the region of convergence for an indirect method is usually smaller than that for a direct method, i.e. the indirect method needs a better initial guess.

Secondly, to compute the gradient, indirect methods use analytic expressions, while direct methods use finite difference approximations. Choosing a forward difference versus central difference method to calculate the gradient information significantly impacts the accuracy of the gradient information ($O(\Delta^2)$ vs $O(\Delta)$) but is also a tradeoff in terms of computational expense.

Thirdly, a big problem with indirect methods is deriving the necessary optimal conditions which involves solving differential equations, path constraints, and boundary conditions, which can be complicated to solve. In turn, this means that current indirect methods are not very flexible.

Betts describes direct shooting and indirect shooting methods in considerable detail.

Direct shooting uses a small number of NLP variables and rolls out the dynamics to achieve constraint satisfaction by solving an initial value problem. This is often used in problems where the number of NLP variables is low. In orbit transfer problems with impulsive burns, usually the thrust variation is not modeled very intricately because most real vehicles don’t have the ability to implement a variable direction thrust even if one was computed. So direct shooting methods using four variables per burn, i.e. the time of ignition and the velocity increment, work perfectly well for problems like this. The performance of a direct shooting method worsens significantly as the number of NLP variables increases.

Indirect shooting methods differ from direct shooting methods because the control is defined at each time step. The biggest problem with these is the sensitivity to the initial guess. Some of Betts solutions to this problem include using the solution to a previously solved similar problem as a starting point for this new problem, assuming the solutions are in the same homotopy class. We can also use the sweep method to integrate state equations forward and adjoint equations backward. As with any indirect method, the NOCs need to be derived and solved explicitly.

When discussing implicit and explicit multi-step integration processes, Betts brings up the interesting point that the trajectory optimization problem is a boundary value problem, not an initial value problem, which means that any scheme is effectively implicit.

Another interesting point that Betts brings up is how in the aerospace field, the input data is often tabular in nature, be it because analytic physics-based representations are not possible, or that the data comes from experiments. This means that the numerical implementation of a trajectory optimization problem needs to be able to deal with tabular data. Linearly interpolating tabular data results in non-differentiability at the data points, leading to difficulties in numerical integration techniques that assume smooth differentiable functions. Betts suggests many alternatives for representing tabular data in a smooth functional form, e.g. using a quadratic drag polar or B-splines.

I didn't know about the methods of index reduction for path constraint functions if the partial derivative of the path constraint with respect to the control is not full rank. I was especially interested in how similar methods can be used if the Hessian with respect to u is a singular matrix, i.e. in the case of what Betts calls a "singular arc".

In summary, Betts also touches upon the main problems with dynamic programming, which is the curse of dimensionality, and genetic algorithms, which is the fact that discrete variables make it hard to determine convergence of a trajectory optimization algorithm and that they do not exploit gradient information.

Betts also hypothesizes that the biggest improvements to current methods will probably come from advances in high-index differential-algebraic equations, automatic differentiation, and sparse nonlinear programming.