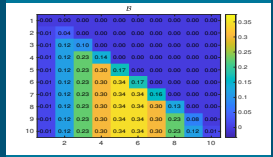
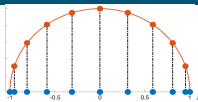
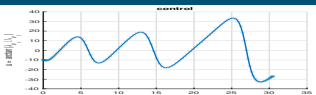
National Laboratories

Spectral Integration for Trajectory Optimization







Aurya Javeed, Drew Kouri, Denis Ridzal (Sandia)
I. Michael Ross (Naval Postgraduate School)

Trajectory Optimization

The focus of our work is trajectory optimization, i.e., ODE-constrained optimization problems of the form

$$\begin{aligned} \underset{u,T}{\text{minimize}} \quad & f(x,u,T) \\ \text{subject to} \quad & 0 = c(x,u,T) \triangleq \begin{cases} \dot{x} - g(x,u,t) \\ x(0) - x_i \end{cases} \end{aligned} \tag{1}$$

In words, the goal is to find an input function u and terminal time T for which the objective f is smallest.

The formulation (1) is without loss of generality. Additional (in)equality constraints can be handled with, e.g., an augmented Lagrangian method like ALESQP [1]. This class of algorithms wrap an "outer loop" around (1) that modifies f.

Discretizing ODEs

Discretizing ODEs

Techniques for discretizing an initial value problem, $\dot{x} = g(x, t)$, $x(0) = x_0$, include

1. Runge-Kutta (RK)

$$x(t_i + h) - x(t_i) \approx h \int_0^1 g\big(x(t_i + sh), t_i + sh\big)\, ds$$

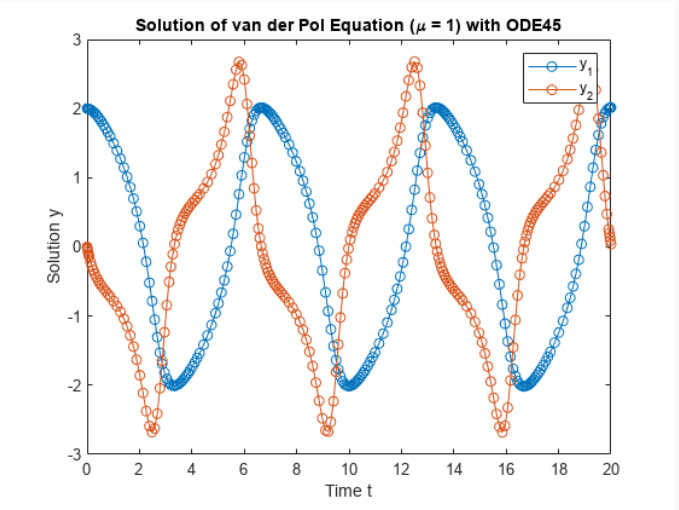
Were g a function of t only, we could approximate the integral with quadrature.

RK methods are quadrature rules with approximations of x at the quadrature points.

Each RK method is specified by a Butcher tableau:

$$\begin{array}{c|ccc} c_1 & & & \\ \vdots & & A & \\ c_n & & & \\ \hline & b_1 & \cdots & b_n \end{array}$$

The c_i in the tableau are the quadrature points and the b_i are the quadrature weights [4].



From MathWorks.

Example:

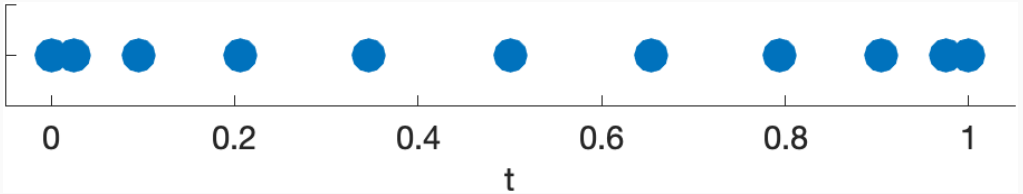
A midpoint rule with an Euler approximation of $x(t_i + \frac{1}{2}h)$.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

Discretizing ODEs

2. Collocation

Let $\vec{c} = (c_1, \ldots, c_n)$ be grid of n times, e.g.,



Collocation is a finite element method. To solve $\dot{x} = g(x, t), x(0) = x_0$, the technique

1. Approximates x as a degree $n-1$ polynomial, $p_n(t) \in \mathbb{P}^n \cong \mathbb{R}^n$.
2. Enforces
   - the initial condition $p_n(0) = x_0$
   - the dynamics $\dot{p}_n = g(p_n, t)$ at $n-1$ of the c_i.

These n conditions define a system of n equations for n unknowns

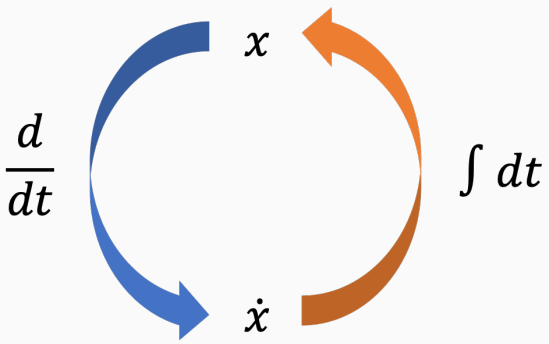
$$\vec{p} \triangleq p_n|_{\vec{c}} = \big(p_n(c_1), \ldots, p_n(c_n)\big),$$

which uniquely specify $p_n(t)$.

The Same... But Different

Collocation is a special instance of RK.

Instead of differentiating the state, however, the RK formulation "goes backward" by approximating the derivative $\dot{x} = g$ with a polynomial and integrating.



We discretize trajectory optimization problems in this way because it scales:

- "Derivative-based" collocation genereates equations that become increasingly *ill-conditioned* as the discretization size $n \to \infty$

$$\left(\text{the operator } \frac{d}{dt} : (C^1, \|\cdot\|_\infty) \to (C^0, \|\cdot\|_\infty) \text{ is unbounded}\right).$$

- "Integration-based" collocation does not have this issue [3].

We are unsure why the former is more popular.

RK Collocation

Newton-based methods solve nonlinear equations using a sequence of linear ones, so – without loss of generality – we consider

$$\begin{cases} \dot{x} = g(t)x \\ x(0) = x_0. \end{cases} \tag{2}$$

Our "RK collocation" discretization of (2) is a linear system

$$(I - BD)\vec{p} = \vec{b}. \tag{3}$$

Here, I is the identity, D is sparse, and B is dense. Concretely,

$$B_{ij} = \int_0^{c_i} \ell_j(s)ds,$$

where ℓ_j is the jth Lagrange polynomial for the collocation points $\vec{c}$.[1]

A surprising result: We can solve (3) in $\mathcal{O}(n \log n)$ time and space (i.e., more cheaply than we can write it down!).

[1] B is in fact the "RK matrix" in the method's Butcher tableau.

Spectral Integration [3]

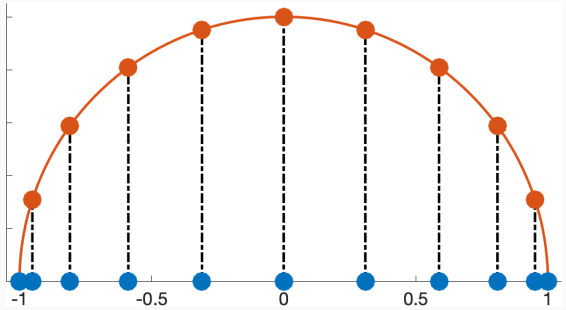
$$(I - BD)\vec{p} = \vec{b} \qquad (3)$$

We collocate at Chebyshev points,

$$c_k = \cos\left(\frac{k-1}{n-1}\pi\right).$$

Doing so allows us to

"apply the integral operator [B (and hence $I - BD$)] in $\mathcal{O}(n \log n)$ operations, making iterative methods more attractive... Unfortunately, for many situations of interest, complex behavior of the solution causes ... the number of iterations to be large" [3].



$$p_n(t) = \sum_{j=0}^{n-1} a_j T_j(t),$$

where T_j is the jth Chebyshev polynomial:

$$T_j(t) = \cos(j \operatorname{arccos}(t)).$$

$B : \mathbb{R}^n \to \mathbb{R}^n$

1. interpolate the values $\vec{v} \in \mathbb{R}^n$ on $\vec{c}$	FFT
2. integrate this interpolant	$\mathcal{O}(n)$
3. evaluate the integral on $\vec{c}$	IFFT

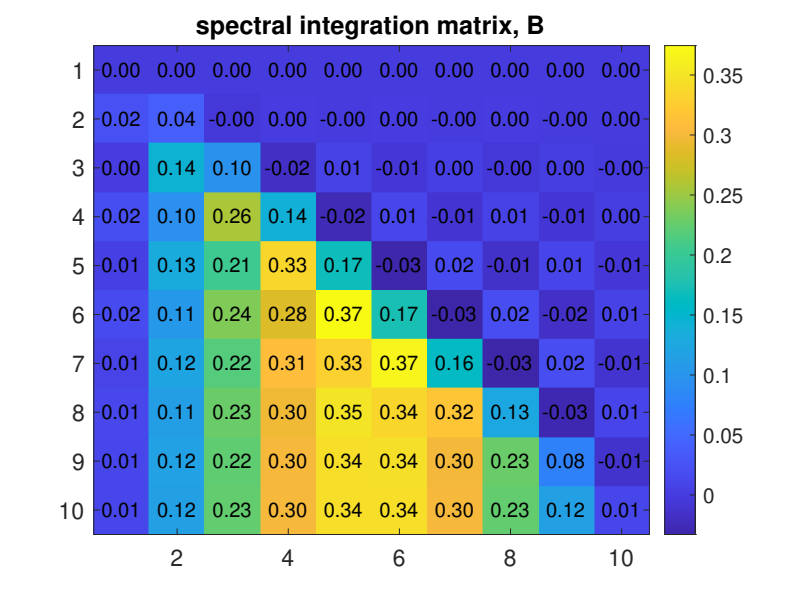
We overcame this issue with a problem-independent preconditioner.

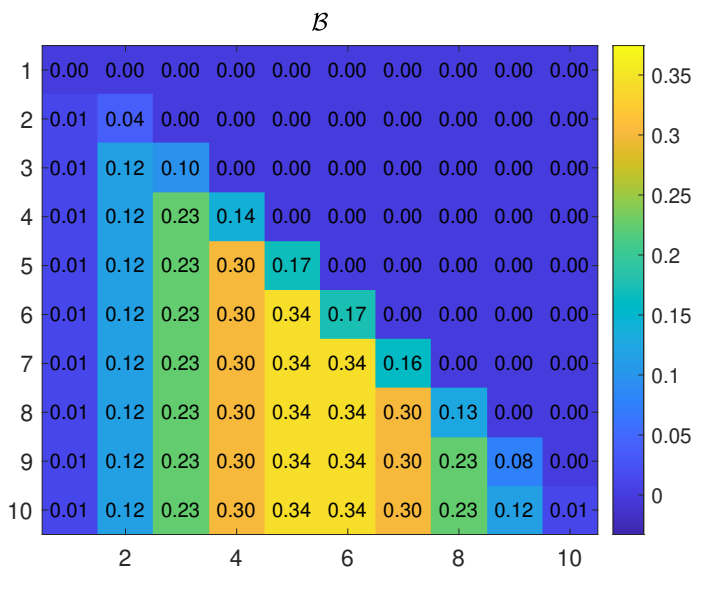
Preconditioning

Preconditioning

As $n \to \infty$, B converges to indefinite integration $\mathscr{I}(\xi)(t) = \int_0^t \xi(s)ds$, which is the continuous analogue of a cumulative sum. Indeed,

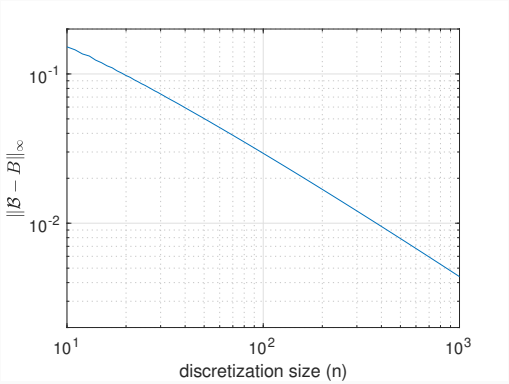
> computing the function $\mathscr{I}(\xi)$ is a matter of continually adding the infinitesimal increment $f(t)ds$ to the running tally $\int_0^t f(s)ds$





The subdiagonal columns of our $\mathcal{B}$ approximation are each constant and equal to quadrature weights. This structure means the preconditioner $\mathcal{P} = I - \mathcal{B}D$ requires only $\mathcal{O}(n)$ operations to invert.

Approximation Accuracy



Sublinear decay of $\|\mathcal{B} - B\|_\infty$.

Theorem 1. (Accuracy of our Preconditioner $\mathcal{P} = I - \mathcal{B}D$)

$\|\mathcal{B} - B\|_\infty \to 0$ at a rate of $\mathcal{O}\left(\frac{\log n}{n}\right)$.

Accuracy of Perturbed Linear Systems [2]

Suppose

$$A\vec{p} = \vec{b} \qquad \text{and} \qquad (A + \Delta A)\vec{q} = \vec{b} + \Delta\vec{b}$$

with $\|\Delta A\| \leq \epsilon\|A\|$ and $\|\Delta\vec{b}\| \leq \epsilon\|\vec{b}\|$.

For sufficiently small ϵ,

$$\frac{\|\vec{q} - \vec{p}\|}{\|\vec{p}\|} \leq \frac{2\gamma}{1 - \gamma}, \quad \text{where} \quad \gamma = \epsilon\kappa(A).$$

Theorem 1 ($\|\mathcal{B} - B\|_\infty \overset{\mathcal{O}(n^{-1}\log n)}{\to} 0$)

Step 1: It suffices to show that $\|U\|_\infty \overset{\mathcal{O}(n^{-1}\log n)}{\to} 0$, where U is the upper triangle of B.

Proof Sketch: The upper triangle of our approximation $\mathcal{B}$ is zero, so the upper triangle of $B - \mathcal{B}$ is the upper triangle of B. The lower triangle of $\mathcal{B} - B$ is the lower triangle of

$$\vec{1}\vec{w}_{cc}^{\,T} - B.$$

Here, $\vec{w}_{cc}$ is the vector of Clenshaw-Curtis quadrature weights,

$$\left(\vec{w}_{cc}\right)_i = \int_{-1}^{+1} \ell_i(t)dt.$$

The symmetry of the Chebyshev nodes and the connection between B and integration gives us

$$\vec{1}\vec{w}_{cc}^{\,T} - B = EBE,$$

where E is the *exchange permutation*

$$E = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

Step 2: $\|U\|_\infty \overset{\mathcal{O}(n^{-1}\log n)}{\to} 0$.

Proof Sketch: We wrote B as a composition of three operations, i.e.,

$$B = V\mathscr{I}C.$$

Submultiplicativity of $\|\cdot\|_\infty$ is too coarse of a bound. We instead work with $V\mathscr{I}C$ directly.

- The product $(V\mathscr{I})C$ includes sums of cosine products that reduce to terms of the form

$$\sum_{k=1}^{n-1} \frac{\sin(\theta k)}{k}. \tag{4}$$

- To bound, (4), we use the complex logarithm:

$$\sum_{k=1}^{\infty} \frac{\sin(\theta k)}{k} = \mathfrak{Im}\left[\sum_{k=1}^{\infty} \frac{\exp(\mathbb{i}\theta)^k}{k}\right] = \mathfrak{Im}\left[-\log\left(1-\exp(\mathbb{i}\theta)\right)\right].$$

Theorem 1 ($\|U\|_\infty \overset{\mathcal{O}(n^{-1}\log n)}{\to} 0$)

- Let

$$\mathcal{S}_n(z) = (1-z)\sum_{k=1}^{n-1}\frac{z^k}{k}$$

so that

$$\left|\sum_{k=1}^{\infty}\frac{\sin(\theta k)}{k} - \sum_{k=1}^{n-1}\frac{\sin(\theta k)}{k}\right| = \left|\mathfrak{Im}\left[\sum_{k=n}^{\infty}\frac{\exp(\mathbb{i}\theta)^k}{k}\right]\right| \leq \frac{\sup_{|z|\leq 1}|\mathcal{S}_\infty(z) - \mathcal{S}_n(z)|}{\left|1-\exp\left(\mathbb{i}\theta\right)\right|}.$$

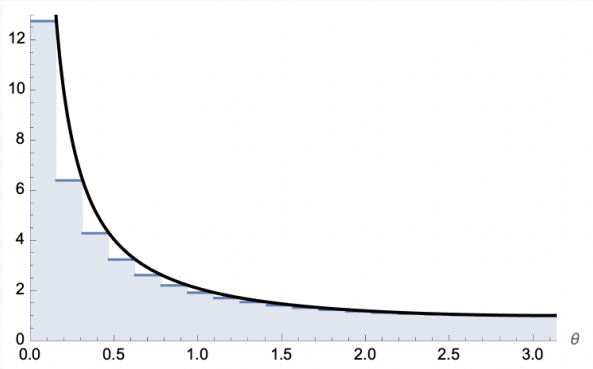
- On the closed unit disk, $\mathcal{S}_n \rightrightarrows \mathcal{S}_\infty$. The final step in showing that

$$\|U\|_\infty \overset{\mathcal{O}(n^{-1}\log n)}{\to} 0$$

at a rate of $\frac{\log n}{n}$ is to therefore bound

$$\left|1-\exp\left(\mathbb{i}\theta\right)\right|^{-1} = \frac{1}{2}\left|\csc\left(\frac{\theta}{2}\right)\right|$$

tightly enough for a sum over the θs that constitutes the ∞ norm.



An integral overapproximation of our cosecant terms.

Numerics

Orbit Transfer

$$\underset{u,T}{\text{minimize}} \quad f(x,u,T)$$

$$\text{subject to} \quad 0 = c(x,u,T) \triangleq \begin{cases} \dot{x} - g(x,u,t) \\ x(0) - x_i \end{cases} \tag{1}$$

We study a version of the orbit transfer problem in [5]:

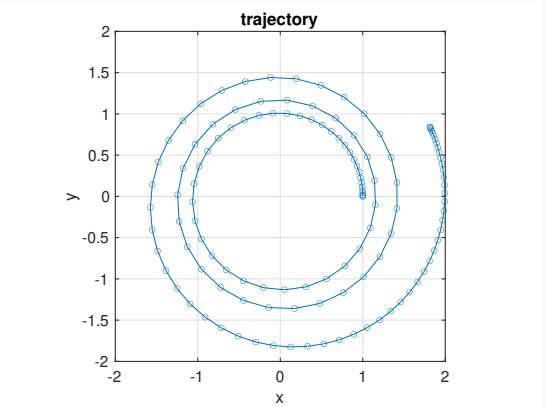
$$f(x,u,T) = T$$

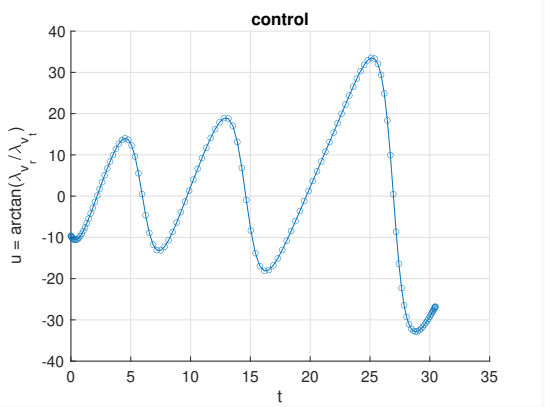
$$x = (r, v_r, v_t)$$

$$g(x,u,t) = \begin{cases} v_r \\ \frac{v_t^2}{r} - \frac{1}{r^2} + a\sin u \\ -\frac{v_r v_t}{r} + a\cos u \end{cases}$$

$$x_i = \left(r_i, 0, r_i^{-1/2}\right)$$

$$x_f = \left(r_f, 0, r_f^{-1/2}\right).$$





The parameter $a > 0$ is the thrust amplitude, which we take to be small.

"Indirect" Methods

The optimality conditions of the continuous problem (1) allow us to eliminate u, thereby reducing the trajectory optimization problem to a nonlinear boundary value problem (BVP).

We reorder unknowns so that the values of the state x at the interior times come first.

$$A\vec{p} = \vec{b} \qquad \text{where} \qquad A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]$$

The A_{11} block here is the "interior" of the $I - BD$ matrix that we can invert quickly.

We can invert A_{11} just as easily as $I - BD$, but we must also contend with the blocks bordering A_{11}. This border is "skinny", so we arrive at a fast BVP inverse by combining our inverse of A_{11} with the Schur complement $A_{22} - A_{21}A_{11}^{-1}A_{12}$.

Linear Solves

We are using GMRES for our fast inverse.

n	unpreconditioned		preconditioned	
	Newton steps	average # of GMRES MVMs per Newton step	Newton steps	average # of GMRES MVMs per Newton step
1024	13	72.38	12	2.17
2048	13	72.92	12	1.92
4096	12	73.75	12	1.92
8192	12	74.25	12	1.92

Solving a nonlinear orbit transfer ODE.

We can continue to, e.g., $n \sim 1{,}000{,}000$ with just a laptop.

For linear ODEs, we have an accurate implicit method at nearly the cost of an explicit method.

Impact

Significance

Returning to trajectory optimization...

$$\begin{aligned} \underset{u,T}{\text{minimize}} \quad & f(x,u,T) \\ \text{subject to} \quad & 0 = c(x,u,T) \triangleq \begin{cases} \dot{x} - g(x,u,t) \\ x(0) - x_i \end{cases} \end{aligned} \tag{1}$$

The key computational kernel of the ALESQP algorithm is linear solves when the matrix has the form

$$M = \begin{pmatrix} R_x & 0 & J^T \\ 0 & R_u & C^T \\ J & C & 0 \end{pmatrix}.$$

J is the only dense block, and it is precisely the $I - BD$ matrix that we can invert in $\mathcal{O}(n \log n)$.

We have an equally fast inverse for J^T and hence an equally fast Schur complement preconditioner for M [6]. This preconditioner, in turn, accelerates ALESQP.

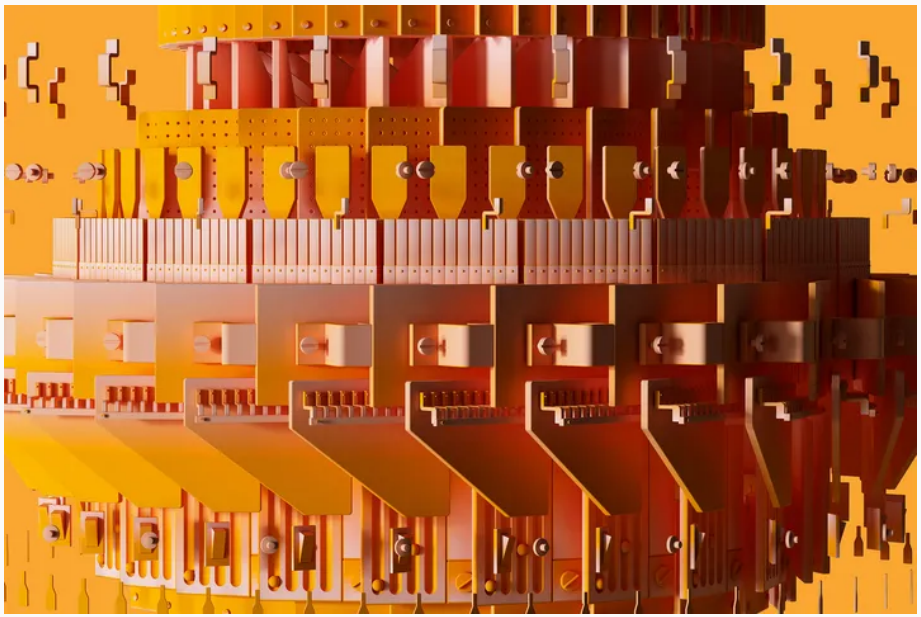
Significance

For edge computing we can do more with less.

To boot,

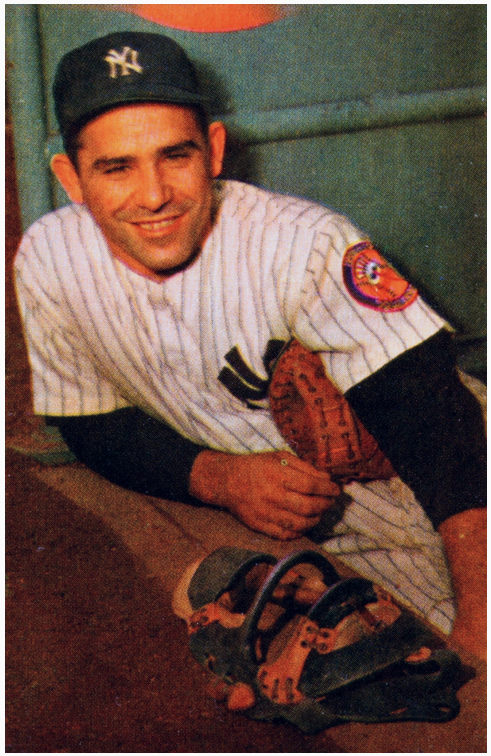
- We require only an FFT library and no direct solvers of any kind.
- Our inverse has helped make new analog hardware viable for trajectory optimization, even at small n.

"An analog computer ... uses the continuous variation aspect of physical phenomena such as electrical, mechanical, or hydraulic quantities ... to model the problem being solved. In contrast, digital computers represent varying quantities ... by discrete values ... " – Wikipedia



From Wired: "The Unbelievable Zombie Comeback of Analog Computing."

Current Work



From the public domain.

"I always thought that record would stand until it was broken."

- Yogi-isms: *"statements that, if taken literally, are ... meaningless ... yet nevertheless convey something true"* [7].

Interpolating a smooth function at Chebyshev points is superalgebraically accurate: the approximation error decays faster than $\mathcal{O}(n^{-m})$ for all m.

- Is this an "inverse Yogi-ism"? Are these global discretizations the way to go for trajectory optimization?

Software

A Pyomo-interfaced ROL toolkit for trajectory optimization.

Theory

h and p error analyses for our spectral integration discretization of trajectory optimization.

Conclusions

- We collocate with integrals instead of derivatives.
- For linear ODEs, we have a $\mathcal{O}(n \log n)$ iterative solve that based on the FFT and a cumulative sum preconditioner.
- We are using this solve to accelerate trajectory optimization.

Thanks!

References

H. Antil, D. P. Kouri, and D. Ridzal.

ALESQP: An augmented Lagrangian equality-constrained SQP method for optimization with general constraints.

SIAM Journal on Optimization, 33(1):237–266, 2023.

G. H. Golub and C. F. Van Loan.

Matrix Computations.

Johns Hopkins University Press, third edition, 2013.

L. Greengard.

Spectral integration and two-point boundary value problems.

SIAM Journal on Numerical Analysis, 28(4):1071–1080, 1991.

E. Hairer, S. P. Nørsett, and G. Wanner.

Solving Ordinary Differential Equations I, Nonstiff Problems.

Springer, 2nd edition, 1993.

N. Koeppen, I. M. Ross, L. C. Wilcox, and R. J. Proulx.

Fast mesh refinement in pseudospectral optimal control.

Journal of Guidance, Control, and Dynamics, 42(4):711–722, 2019.

T. Rees, H. S. Dollar, and A. J. Wathen.

Optimal solvers for PDE-constrained optimization.

SIAM Journal on Scientific Computing, 32(1):271–298, 2010.

L. Trefethen.

Inverse Yogiisms.

Notices of the American Mathematical Society, 63(11), 2016.