Discretizing Constraints at Multiple Scales

Nicholas Richardson

11 April 2025

THE UNIVERSITY OF BRITISH COLUMBIA
Mathematics
Faculty of Science

INSTITUTE OF APPLIED MATHEMATICS
University of British Columbia

A Simple Question…

Given $x = (x_1, x_2, \ldots, x_I) \in \mathbb{R}^I$ with entry sum $S \in \mathbb{R}$

$$\sum_{i=1}^{I} x_i = S,$$

what does the subvector $\overline{x} = (x_1, x_3, \ldots, x_{I-1}) \in \mathbb{R}^{I/2}$ sum to

$$\sum_{j=1}^{I/2} x_{2j-1} = \ ?$$

Answer

No trick here, the subvector could sum to anything.

But what if x comes from a discretization of a (Lipschitz) continuous function

$$x_i = f(t_i).$$

$$\sum_{i=1}^{I} x_i = S \implies \sum_{j=1}^{I/2} x_{2j-1} = \ ?$$

Where does this come from?

Optimize over smooth probability density functions

$$\min_{p \in C^1([a,b])} \ell(p) \quad \text{s.t.} \quad \int_a^b p(t)dt = 1 \text{ and } p(t) \geq 0$$

by discretizing

$$\min_{x \in \mathbb{R}^I} \ell(x) \quad \text{s.t.} \quad \|x\|_1 = 1 \text{ and } x \geq 0$$

where coordinates $x = (x_1, x_2, \ldots, x_I)$ are defined as

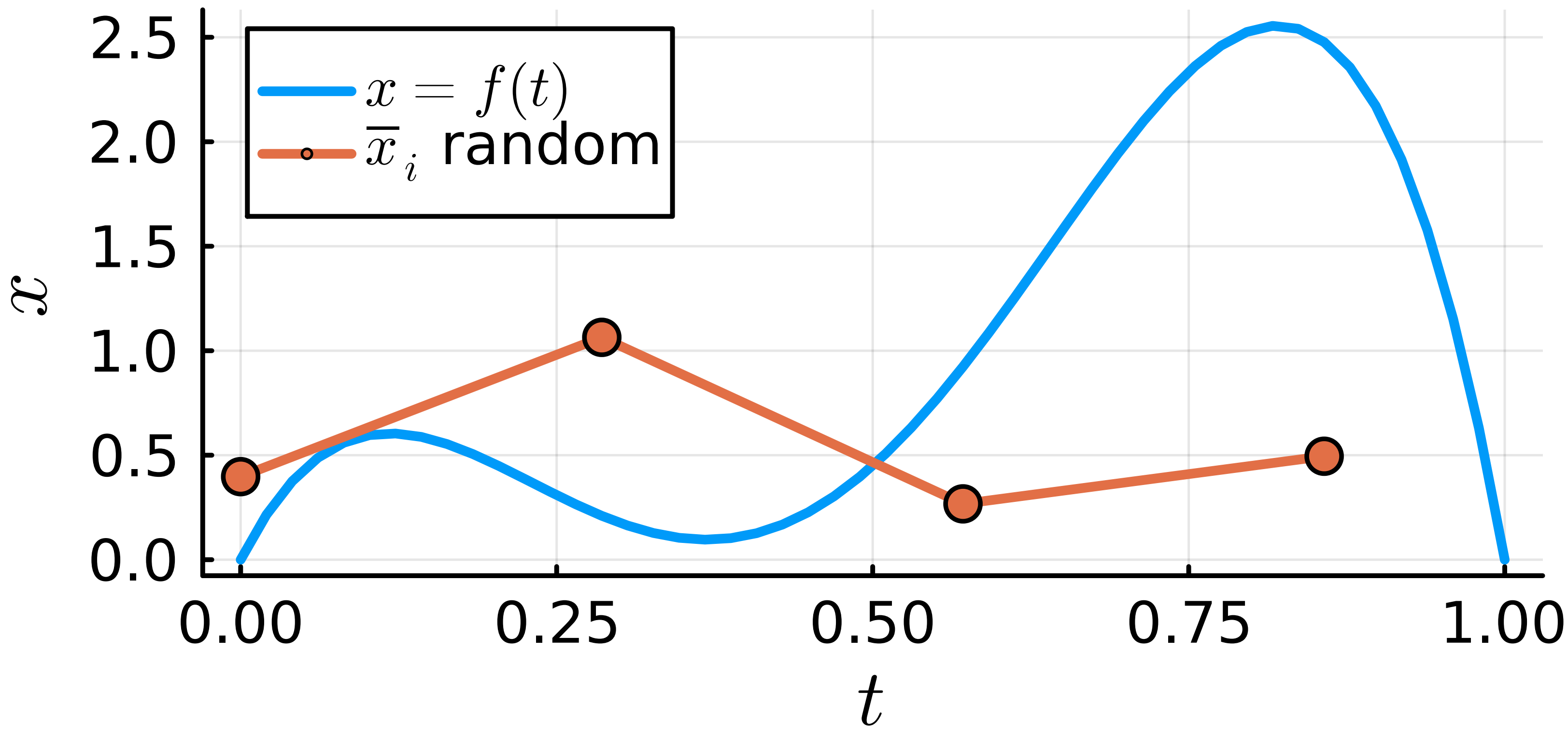
$$x_i = p(t_i)\Delta t.$$

Multi-scaled optimization

- Rather than optimize over all entries x_i
- optimize every other entry first
- then interpolate the result
- use the interpolate solution as a better initialization to the larger problem

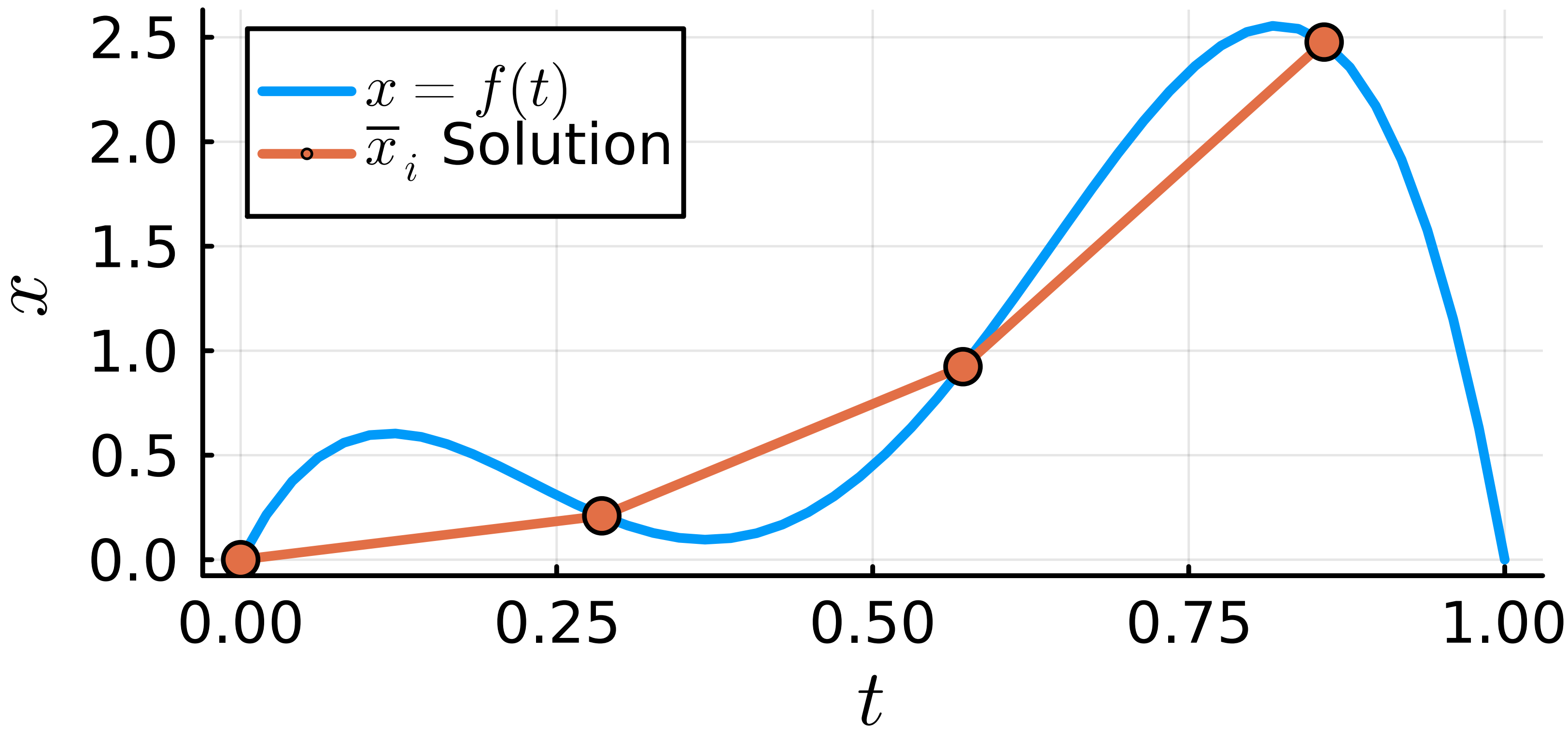
Example Plot

```
f(t) = -84t^4 + 146.4t^3 - 74.4t^2 + 12t
I = 8
t = range(0, 1, length=I)
x = f.(t)
```



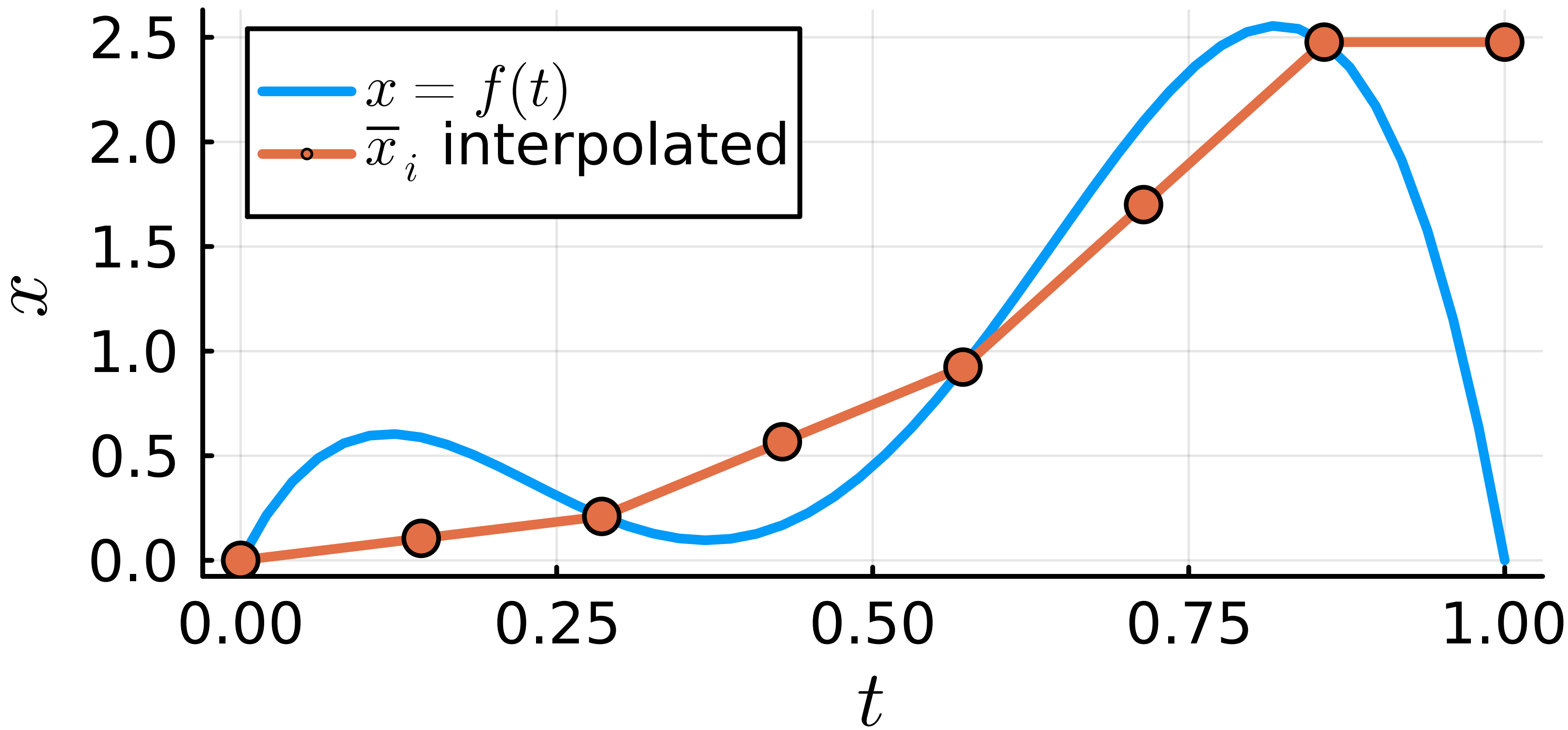
Example Plot

```
f(t) = -84t^4 + 146.4t^3 - 74.4t^2 + 12t
I = 8
t = range(0, 1, length=I)
x = f.(t)
```



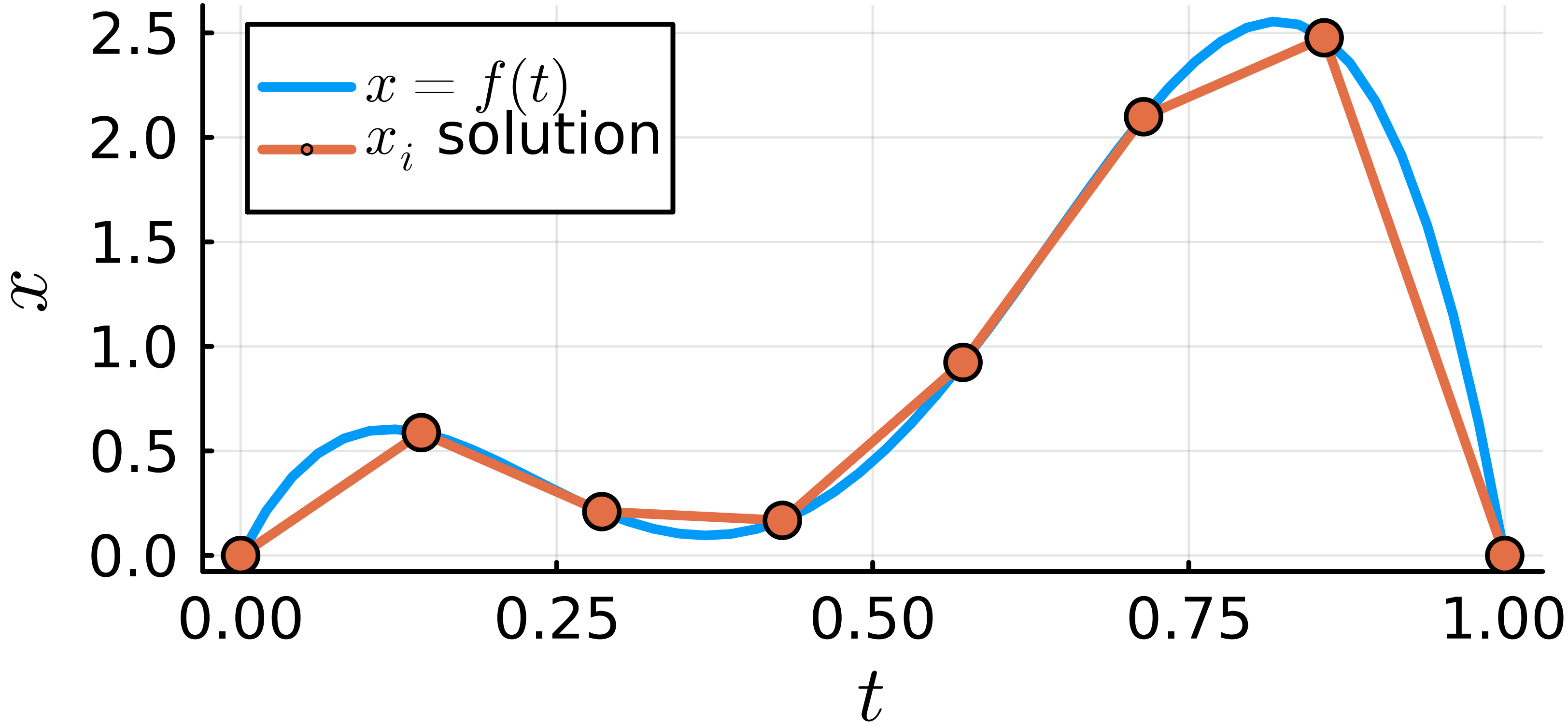
Example Plot

```
f(t) = -84t^4 + 146.4t^3 - 74.4t^2 + 12t
I = 8
t = range(0, 1, length=I)
x = f.(t)
```



Example Plot

```
f(t) = -84t^4 + 146.4t^3 - 74.4t^2 + 12t
I = 8
t = range(0, 1, length=I)
x = f.(t)
```



Advantages

- Can start on a super coarse grid (as small as 2 points!)
- Spend most of the work solving smaller problems
- But if $\|x\|_1 = 1$, what constraint to use on smaller problems?

Back to the question

For *any* vector $x \in \mathbb{R}^I$, if $\|x\|_1 = S$, you don't know what the subvector $\overline{x}$ sums to.

But if x comes from a uniform discretization of an L-Lipschitz function f, then $\overline{x}$ sums to $S/2$ (at least close to it)!

$$\left| \frac{S}{2} - \|\overline{x}\|_1 \right| \leq \frac{I}{I-1} \frac{L(b-a)}{4}$$

Normalize $z = x/\|x\|_1$, then $\|\overline{z}\|_1 \approx 1/2$

$$\left| \frac{1}{2} - \|\overline{z}\|_1 \right| \leq \frac{I}{I-1} \frac{L(b-a)}{4S}$$

Numerical Example

```
# Make grid
I = 2^6
t = range(0, 1, length=I)

# Sample function and normalize
f(t) = -84t^4 + 146.4t^3 - 74.4t^2 + 12t
x = f.(t)
S = sum(abs, x)
x ./= S # normalize x
x_subsampled = @view x[begin:2:end]

theory_error = I/(I-1) * 33.6 * (1-0) / 4S
actual_error = abs(1/2 - sum(abs, x_subsampled))
```

```
Number of points: 64
Expected Sum: 0.50000          Theoretical error: 0.13558
  Actual Sum: 0.50068             Observed error: 0.00068
```

Worst Case Example

```
# Make grid
I = 2^6
t = range(0, 1, length=I)

# Sample function and normalize
f(t) = t
x = f.(t)
S = sum(abs, x)
x ./= S # normalize x
x_subsampled = @view x[begin:2:end]

theory_error = I/(I-1) * 1 * (1-0) / 4S
actual_error = abs(1/2 - sum(abs, x_subsampled))
```

```
Number of points: 64
Expected Sum: 0.50000          Theoretical error: 0.00794
  Actual Sum: 0.49206             Observed error: 0.00794
```

Extensions

- Linear constraints more generally $\langle a, x \rangle = b$?
  - can prove similar bounds for $\langle \overline{a}, \overline{x} \rangle \approx b/2$
- Other p-norms $\|x\|_p = 1$?
  - conjecture $\|\overline{x}\|_p \approx (1/2)^{1/p}$
- What's the best scheme for optimizing over many scales?