Bounded Analog Complexity

When wall-clock time becomes a faithful resource

Xiang Huang

joint work with Ho-Lin Chen

University of Illinois Springfield National Taiwan University

DNA32 · Fayetteville · August 3, 2026

arXiv:2607.12234



MY WIFE

“How long does the air show last?”

ME

“Just measure the length of the airplanes’ trajectories.”

MY WIFE

“Isn’t there something called a clock?”

Exactly. Why does analog complexity measure trajectory length instead of time?

Review: an analog computer is a PIVP

POLYNOMIAL INITIAL VALUE PROBLEM

$$y'(t) = p(y(t)), \qquad y(0) = y_0 \in \mathbb{Q}^d,$$

where $p : \mathbb{R}^d \to \mathbb{R}^d$ is a vector of polynomials with rational coefficients.

A designated coordinate is the output:

$$f(t) = y_1(t).$$

A minimal example

$$x' = 1 - x, \qquad x(0) = 0.$$

Its exact trajectory is

$$x(t) = 1 - e^{-t} \longrightarrow 1.$$

To obtain error $< e^{-r}$, wait $t > r$.

The evolving trajectory $y(t)$ is the analog computation.

Composition lets the clock lie

GPAC-generable functions are closed under composition. If $f(u) \to \alpha$, we may feed it a different PIVP clock.

Exponential accelerator

$$u' = 1 + u, \quad u(0) = 0 \quad \Longrightarrow \quad u(t) = e^t - 1.$$

Run $f(u(t))$ instead of $f(t)$:

$$\mu_{\text{new}}(r) = \log\bigl(1 + \mu_f(r)\bigr).$$

Finite-deadline accelerator

$$v' = (1+v)^2, \quad v(0) = 0 \quad \Longrightarrow \quad v(t) = \frac{t}{1-t}.$$

As $t \to 1^-$, $v(t) \to \infty$:

$$f(v(t)) \longrightarrow \alpha.$$

An infinite internal computation can be packed into $0 \le t < 1$. The raw parameter t is not a meaningful cost.

Bournez–Graça–Pouly: charge for distance traveled

Measure the length of the solution curve, not the clock parameter:

$$\boxed{L_y(T) = \int_0^T \|y'(t)\| \, \mathrm{d}t.}$$

For any increasing clock g with $g(0) = 0$, and $z(t) = y(g(t))$,

$$\begin{aligned} L_z(T) &= \int_0^T \|y'(g(t))\| \, g'(t) \, \mathrm{d}t \\ &= \int_{g(0)}^{g(T)} \|y'(u)\| \, \mathrm{d}u = L_y(g(T)). \end{aligned}$$

Reparametrization changes speed, but it does not shorten the trajectory.

BGP framework

Polynomial trajectory length characterizes polynomial-time computable functions.

A CRN is a special GPAC

then carry the BGP framework into chemistry

MASS-ACTION FORM

A reaction $\nu_r \to \nu_r'$ fires at rate $k_r x^{\nu_r}$. Hence

$$\dot{x}_i = \sum_r k_r(\nu'_{r,i} - \nu_{r,i})x^{\nu_r}.$$

Split production from consumption:

$$\dot{x}_i = P_i(x) - x_i Q_i(x), \qquad P_i, Q_i \in \mathbb{R}_{\geq 0}[x].$$

Every negative term consumes X_i, so it must carry the factor x_i.

FOLLOW THE BGP ROUTE

This is a polynomial ODE, but with the kinetic sign constraint

$$x_i = 0 \implies \dot{x}_i = P_i(x) \geq 0.$$

Applying the BGP framework imports:

1. trajectory length as cost;
2. no a priori bound on concentrations.

Mass-action CRNs are constrained PIVPs. Thus BGP applies—and imports its unbounded ambient model.

Our viewpoint: realistic chemistry is bounded

BOUNDED SPECIES

Every concentration stays below some fixed limit.

↓

The whole system has a maximum speed.

↓

$$\text{arc length by physical time } t \;\leq\; \text{fixed constant} \times t$$

With bounded species, physical time becomes meaningful again.

2017: bounded CRNs compute real numbers

Xiang Huang, Titus H. Klinge, James I. Lathrop, Xiaoyuan Li, and Jack H. Lutz

A CRN computes α through the limit of one designated species X.

ZERO INITIALIZATION

Every species begins at zero:

$$x(0) = 0.$$

BOUNDED SPECIES

Every concentration remains bounded for all physical time.

REAL-TIME READOUT

For every $t \geq 1$,

$$\left|X(t) - |\alpha|\right| \leq 2^{-t}.$$

Integral rate constants make the CRN finitely specifiable.

UCNC 2017 · Fayetteville, Arkansas

My first academic talk

That was where this line of work began for me.

Nine years later, I am back in Fayetteville.

What complexity was hiding inside that model?

Then the line continued

2019 · DNA 25

Xiang Huang, Titus H. Klinge, and James I. Lathrop

Same computability

Bounded CRNs and bounded GPACs compute exactly the same real numbers.

2022 · DNA 28

Xiang Huang and Rachel Huls

Same computability again

Large-population protocols, bounded CRNs, and bounded GPACs compute exactly the same real numbers.

Across this line of work, boundedness was the common premise.

Before today: boundedness was simply required

A HARD RULE

Everything in the system had to stay bounded.

We assumed this before we started.

If a system did not meet this rule,
we did not study it.

Today: compile away the assumption

Main theorem

Every real number computable by a possibly unbounded PIVP is computable by a bounded PIVP.

Unbounded input

$$y' = p(y), \qquad y_1(t) \to \alpha$$

Unbounded species allowed.



Bounded output

$$\widetilde{y}' = \widetilde{p}(\widetilde{y}), \qquad \widetilde{y}_1(t) \to \alpha$$

Every coordinate is bounded.

Every unbounded species used along the way can be compiled away.

Proof of the main theorem: find every f

A polynomial system with one unbounded species f

$$\begin{aligned}\dot{z}_1 &= a_0(z) + a_1(z)f + a_4(z)f^4,\\ \dot{z}_2 &= b_0(z) + b_2(z)f^2,\\ \dot{z}_3 &= c_0(z),\\ \dot{f} &= d_0(z) + d_3(z)f^3.\end{aligned}$$

Pull out only the f-terms

$$a_1(z)f, \qquad a_4(z)f^4,$$
$$b_2(z)f^2, \qquad d_3(z)f^3.$$

Every one has the same shape:

$$\boxed{g_m(z)f^m}.$$

Here $g_m(z)$ contains no f.

Understand how to eliminate one $g_m(z)f^m$; then apply the same move to every occurrence.

Proof of the main theorem: cancel one term

1 ISOLATE

$$\dot{z}_j = \cdots + \underbrace{g_m(z)\, f^m}_{\text{unbounded factor}} + \cdots$$

2 MAKE A WISH

After $t = s(\tau)$, the chain rule gives

$$\bar{z}_j' = \dot{z}_j(s(\tau)) \underbrace{s'(\tau)}_{\text{our choice}}.$$

We want $\bar{f}^m s'(\tau)$ to stay bounded.

3 CHOOSE THE CLOCK

Use a reciprocal of the same order:

$$s'(\tau) = \frac{1}{1 + \bar{f}(\tau)^m}.$$

Then the offending factor becomes

$$\bar{f}^m s' = \frac{\bar{f}^m}{1 + \bar{f}^m} \in [0, 1).$$

Why the $+1$?
It prevents division by zero when $f = 0$.

Why does the new time reach infinity?

$$\tau(t) = \int_0^t \left(1 + f(u)^m\right) \mathrm{d}u \;\geq\; t \longrightarrow \infty.$$

Proof of the main theorem: make it polynomial

THE NEW PROBLEM

$$U_m = \frac{\bar{f}^m}{1 + \bar{f}^n}$$

It is bounded, but it is still a **rational function**.

A PIVP needs a polynomial ODE.

$\Longrightarrow$

THE SECOND TRICK

Give every ratio its own variable:

$$U_0,\ U_1,\ \ldots,\ U_n.$$

Differentiate the identities together. Each U_m' uses only the other U's and the remaining bounded variables.

The variables update one another recursively.

No division remains.

The finite U-family closes as a polynomial ODE. Then remove f and repeat.

What the proof preserves

SAME ANSWER

The compiler changes the clock, not the destination.

$$\text{original system} \longrightarrow \alpha,$$
$$\text{compiled system} \longrightarrow \alpha.$$

Both compute the same real number.

POLYNOMIAL IN, POLYNOMIAL OUT

Before compilation
polynomial arc-length cost

↓

COMPILER

↓

After compilation
polynomial arc-length cost

Same answer. Polynomial before; polynomial after.

Example: computing a pole

OLD TIME t

$$x' = 1 + x^2, \qquad x(0) = 0,$$

so $x(t) = \tan t$.

The pole occurs at old time $\boxed{t = \dfrac{\pi}{2}}$.

NEW TIME τ

Store the old time in another state variable:

$$s'(\tau) = U_0(\tau), \qquad s(0) = 0,$$

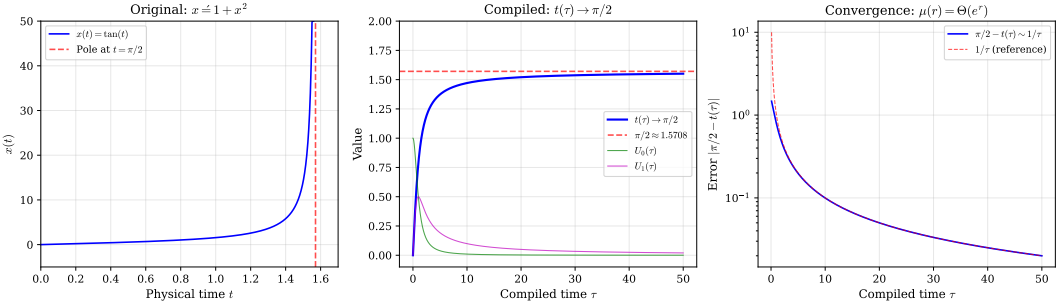
$$s(\tau) = \int_0^{\tau} U_0(\sigma)\, \mathrm{d}\sigma.$$

The full integral is exactly $\pi/2$. We do not need to follow its details here.

new time $\tau \longrightarrow \infty$ $\Longrightarrow$ recorded old time $s(\tau) \longrightarrow \dfrac{\pi}{2}$

Tradeoff: this generic construction has $\mu(r) = \Theta(e^r)$.

Example: the pole becomes a bounded limit



Old time stops at $\pi/2$. New time runs forever while the recorded old time converges to $\pi/2$.

Now that time is fixed, we can define complexity

WE HAVE NAILED DOWN PHYSICAL TIME

How long must we wait for the answer to stay accurate?

Request accuracy r

We want the error below

$$e^{-r}.$$

Larger r means greater accuracy.

$\Longrightarrow$

Measure the waiting time

$\mu(r)$ is the physical time after which the output stays that accurate:

$$t \geq \mu(r) \quad \Longrightarrow \quad |x(t) - \alpha| < e^{-r}.$$

Complexity is how $\mu(r)$ grows. Next: fix the answer 1 and vary only the system.

One simple answer, many complexity classes

Nothing is special about 1. We fix this simple target and vary only the bounded ODE system.

precision clock $g(t)$	output error $1 - x(t)$	modulus $\mu(r)$
t	e^{-t}	$\Theta(r)$
$e^{W(t)} - 1$	$e^{1-e^{W(t)}}$	$\Theta(r \log r)$
$(1+t)^{1/n} - 1$	$e^{1-(1+t)^{1/n}}$	$\Theta(r^n)$
$\log^{(k+1)} t$	$e^{-\log^{(k+1)} t}$	$\Theta(\exp^{(k+1)} r)$

The second row is an $r \log r$ example: complexity belongs to the system, not to 1.

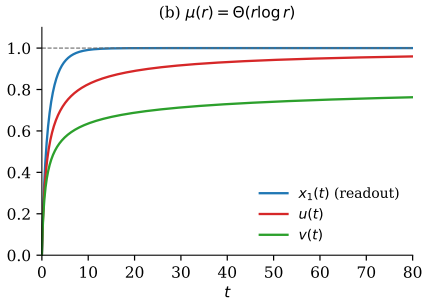
One concrete system: $r \log r$ time

An explicit bounded polynomial ODE

$$u' = (1-u)^2(1-v),$$
$$v' = (1-v)^3(1-u),$$
$$x' = (1-x)(1-v).$$
$$u(0) = v(0) = x(0) = 0.$$

Every variable stays bounded.

The output satisfies $x(t) \to 1$.



To reach error below e^{-r}, wait

$$t \geq (r+1)\log(r+1).$$

$\mu(r) = \Theta(r \log r)$

Wall-clock time sees what arc length forgets

EVERY SYSTEM WE JUST DISCUSSED COMPUTES 1

What the wall clock sees

linear time

$r \log r$ time

r^n time

iterated-exponential time

Different running times



What arc length sees

Every trajectory travels only a finite total distance.

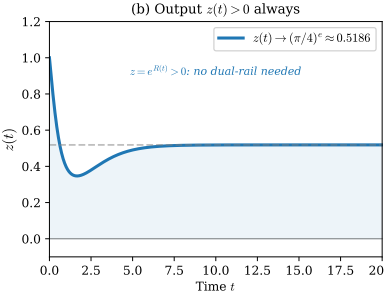
constant

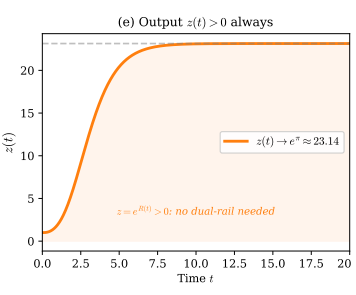
No complexity information

Wall-clock time preserves the differences. Arc length erases them.

Examples: $(\pi/4)^e$ and e^π

If bounded GPACs compute $\alpha > 0$ and β, then a bounded GPAC computes α^β.





Both computations inherit linear time from their inputs.

From a bounded GPAC to chemistry



Dual rail is exact

Signed state x is represented by

$$x = X^+ - X^-,$$

and the difference follows the original ODE.

No severe slowdown

The readout follows the incoming value toward its limit.
The exact speed depends on how quickly that incoming value settles.
A polynomial computation remains polynomial.

The readout may slow things down, but never catastrophically.

Question: are bounded rational ODEs Turing complete?

THE SYSTEM

A finite rational description

$$y' = p(y)$$

- p has rational coefficients;
- the initial values are rational;
- every variable stays bounded for all time.

?

THE QUESTION

Can this bounded system simulate every Turing machine?

In plain language: can it reproduce arbitrarily long discrete computations while *every* continuous variable remains bounded?

Bounded state. Rational data. Unbounded computation?

Take-home messages

1. **Boundedness makes wall-clock time meaningful.** No diverging state can hide infinite work in finite physical time.
2. **Boundedness does not reduce real-number computability.** Surrogate compilation replaces growth by bounded polynomial dynamics while preserving limits and polynomial cost.
3. **Physical time reveals a rich analog hierarchy.** Fine-grained moduli survive exponentiation and the full CRN readout pipeline.

Thank you

ACKNOWLEDGEMENTS

We gratefully acknowledge support from

Ho-Lin Chen

NSTC 國家科學及技術委員會
National Science and Technology Council

National Science and Technology Council (Taiwan)

Grant 113-2221-E-002-204-MY3

Xiang Huang

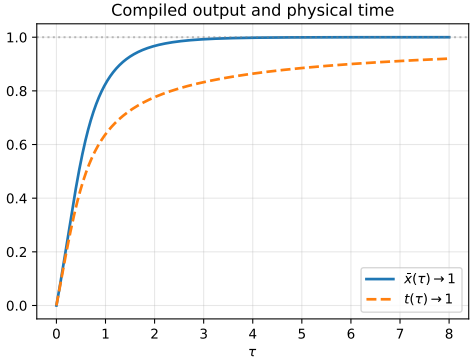
U.S. DEPARTMENT OF ENERGY | Office of Science

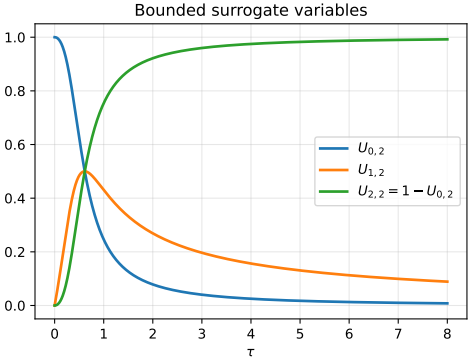
U.S. Department of Energy
Office of Science, ASCR

Grant DE-SC0024278

arXiv:2607.12234 · xianghuang.org

Backup: the finite-time explosion after compilation





Original: $v(t) = 1/(1-t) - 1 \to \infty$

Compiled: $U_0, U_1, U_2, \bar{x} \in [0, 1]$

Backup: the full one-coordinate compiler

For $p_j(z,f) = \sum_m g_{j,m}(z) f^m$ and $p_f(z,f) = \sum_m h_m(z) f^m$, define

$$U_m = \frac{\bar{f}^m}{1 + \bar{f}^n}, \qquad H(\bar{z}, U) = \sum_m h_m(\bar{z}) U_m.$$

The compiled PIVP is

$$\boxed{\begin{aligned} \bar{z}_j' &= \sum_{m=0}^{n} g_{j,m}(\bar{z}) U_m, \\ U_m' &= \big(m U_{m-1} - n U_{n-1} U_m\big) H(\bar{z}, U). \end{aligned}}$$

The identity $U_0 + U_n = 1$ is invariant, and the original unbounded coordinate does not occur in the compiled state.

Backup: an explicit $\Theta(r^n)$ bounded family

For every $n \geq 1$,

$$u' = \frac{1}{n}(1-u)^{n+1},$$
$$x' = \frac{1}{n}(1-x)(1-u)^{n-1}, \qquad u(0) = x(0) = 0.$$

Its exact solution satisfies

$$1 - u(t) = (1+t)^{-1/n}, \qquad 1 - x(t) = \exp\bigl(1 - (1+t)^{1/n}\bigr).$$

Hence

$$|1 - x(t)| < e^{-r} \quad \Longleftrightarrow \quad t > (r+1)^n - 1,$$

so $\mu_n(r) = \Theta(r^n)$.

Backup: the $\Theta(r \log r)$ Lambert-W system

With $W(t)e^{W(t)} = t$, use two bounded surrogates

$$u = 1 - e^{-W(t)}, \qquad v = \frac{W(t)}{1 + W(t)}.$$

They close polynomially with the output x:

$$\boxed{\begin{aligned} u' &= (1-u)^2(1-v), \\ v' &= (1-v)^3(1-u), \\ x' &= (1-x)(1-v). \end{aligned}}$$

Since $1 - x(t) = e^{1-e^{W(t)}}$, precision e^{-r} is reached when

$$t \geq (r+1)\log(r+1) = \Theta(r \log r).$$

Backup: exponentiation in one polynomial system

Track $x_1 \to \alpha - 1$, $u = \log(1 + x_1)$, and $v = x_1/(1 + x_1)$:

$$\boxed{\begin{aligned} x_1' &= (x - 1) - x_1, \\ u' &= (1 - v)x_1', \\ v' &= (1 - v)^2 x_1', \\ z' &= z\big(y'u + y(1 - v)x_1'\big), \end{aligned}} \qquad x_1(0) = u(0) = v(0) = 0, \quad z(0) = 1.$$

Then $z(t) = e^{y(t)u(t)} \to \alpha^\beta$. The output stays strictly positive, so selective dual railing can leave it as a single CRN species.

Backup: the exponentiation trajectories

