PRESENTER
Xiaoyuan Li , Ph.D. student.

How to Make Your Soup Compute π *

Xiang Huang[†], Titus H. Klinge[‡], James I. Lathrop[†]
[†]Iowa State University [‡]Drake University

This research was supported in part by National Science Foundation grants 1545028 and 1900716.

Overview

We investigate the class of real numbers that are computable in real time by chemical reaction networks (CRNs), and its relationship to general purpose analog computers (GPAC).
We denote
- $\mathbb{R}_{RTCRN}$="numbers that can be computed by CRNs **fast**,"
- $\mathbb{R}_{RTGPAC}$= "Same thing but by GPACs."

Take-home fact: **they are actually the same thing!**

Today's results:

1. $\mathbb{R}_{RTCRN}$ is a field:
   - You can do addition, **subtraction**, multiplication, and division.
2. $\mathbb{R}_{RTCRN} = \mathbb{R}_{RTGPAC}$:
   - One can implement an algorithm by a GPAC, which is (much) simpler in form. Then compile it into a CRN.
   - It greatly simplifies the proof that a number $\alpha \in \mathbb{R}_{RTCRN}$.
3. $e,\ \pi \in \mathbb{R}_{RTCRN}$: the field $\mathbb{R}_{RTCRN}$ is not boring.

CRNs and GPACs

A CRN is an abstract **mathematical model** of how chemicals interact in a well-mixed volume.

Example: abstract away concrete molecules

$$\begin{array}{c} 3N_2H_4 \to 4NH_3 + N_2 \\ N_2H_4 \to N_2 + 2H_2 \\ 4NH_3 + N_2H_4 \to 3N_2 + 8H_2 \\ 2H_2 + O_2 \to 2H_20 \end{array} \implies \begin{array}{c} 3A \to 4B + C \\ A \to C + 2D \\ 4B + A \to 3C + 8D \\ 2D + E \to 2F \end{array}$$

The General Purpose Analog Computer (GPAC) is an analog computer model first introduced in 1941 by Claude Shannon.

We can view both GPACs and CRNs as **polynomial initial value problems (PIVPs)**:

GPACs:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = p_i(x_1, \cdots, x_n),$$
for $i = 1, 2, \cdots, n$,
where p_i's are polynomials.

CRNs:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = p_i(x_1, \cdots, x_n) - q_i(x_1, \cdots, x_n)x_i,$$
for $i = 1, 2, \cdots, n$,
where p_i's and q_i's are polynomials
with **positive** coefficients.

Negative terms of derivative of x_i must have x_i in it!

Real-time Computability

By the previous definition, CRNs are restricted GPACs. In terms of Ordinary Differential Equations (ODEs), GPACs are more expressive that CRNs.

In the following example, the right-hand side system can not be implemented by CRNs, since the (negative) term in the red box does not have y in it.

Example:

GPACs/ODE:

$$x' = 1 - x.$$

CRNs/ODE:

$$x' = 1 - x,$$
$$y' = 1 - x.$$

We view a variable as an analog signal, or a function of time. Having the ODE system describing a CRN/GPAC, we can analyze the long-term behavior of the system. We say a CRN/GPAC computes a number α, if there is a variable x converging to α **in the limit**. In particular, x converges to α in real time. That is, **exponentially fast**.

Formally, $\alpha \in \mathbb{R}$ is **real-time CRN-computable** (resp., **GPAC-computable**), and we write $\alpha \in \mathbb{R}_{RTCRN}$ (resp., $\alpha \in \mathbb{R}_{RTGPAC}$), if there exist an (**integral**) PIVP induced by a CRN (resp. GPAC) and a variable $x(t)$ in the system with the following properties:
- (**zero initial values**). For all variable $y(t)$, $y(0) = 0$. (This can be relaxed to integral values.)
- (**boundedness**). All variables are bounded by some constant M,
$$y(t) \le M.$$
- (**real-time convergence**).
$$|x(t) - |\alpha|| \le O(2^{-t}).$$

Simulating a GPACs with a CRN

We souped-up the following theorem to simulate GPACs with CRNs.

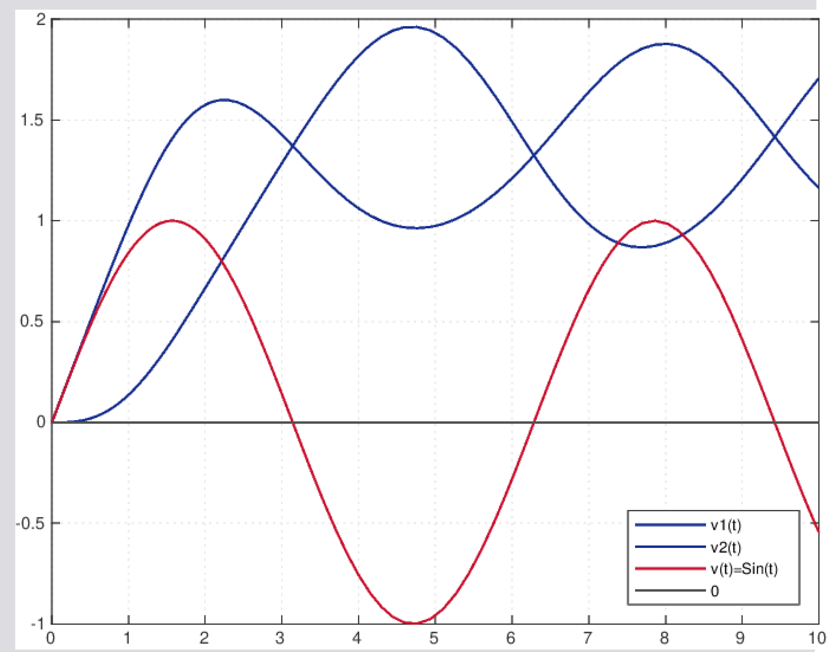
Theorem.(Difference Encoding)
Given a GPAC $\mathcal{A}$, there is a CRN $\mathcal{N}$ that for each variable x in $\mathcal{A}$, there are two variables x_1 and x_2 in $\mathcal{N}$ such that
$$x(t) = x_1(t) - x_2(t) \quad \textbf{pointwise for every t} \ge \textbf{0}.$$

We use a similar construction. Moreover, we also show that **if $x(t)$ is bounded, then x_1 and x_2 are also bounded**.
GPACs are unusually much simpler in form than CRNs. We can first implement a GPAC to compute α, then **compile** it into a CRN. This can be done **automatically**.

A General Procedure

Example:
The function $v(t) = \sin(t)$ can be implemented by the first GPAC/ODE, and encoded by $v_1(t) - v_2(t)$ in the second CRN/ODE.

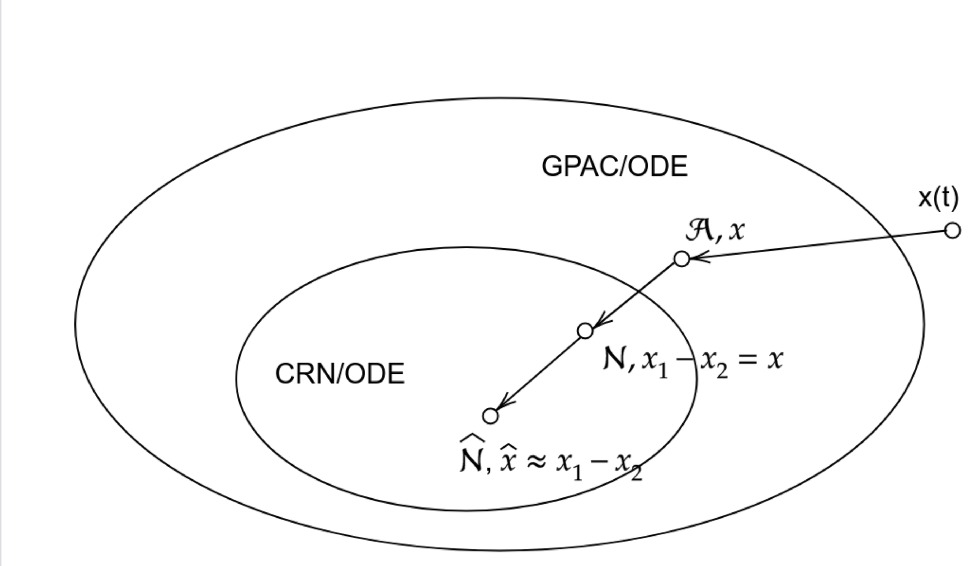
$$\begin{array}{l} u' = v \\ v' = 1 - u \end{array} \implies \begin{array}{l} u_1' = v_1 - u_1u_2 \\ u_2' = v_2 - u_1u_2 \\ v_1' = 1 + u_2 - v_1v_2 \\ v_2' = u_1 - v_1v_2 \end{array}$$



We now can encode every variable x in a GPAC with the difference of **two** variables x_1, x_2 in CRN. However, our definition require we use only **one** variable the chase a number α. We developed a **subtraction** algorithm to make this happen.

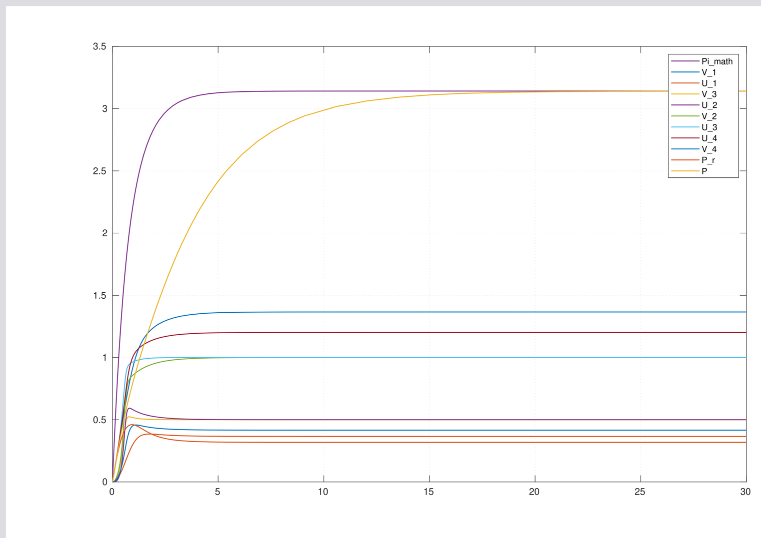
Big picture of the procedure:

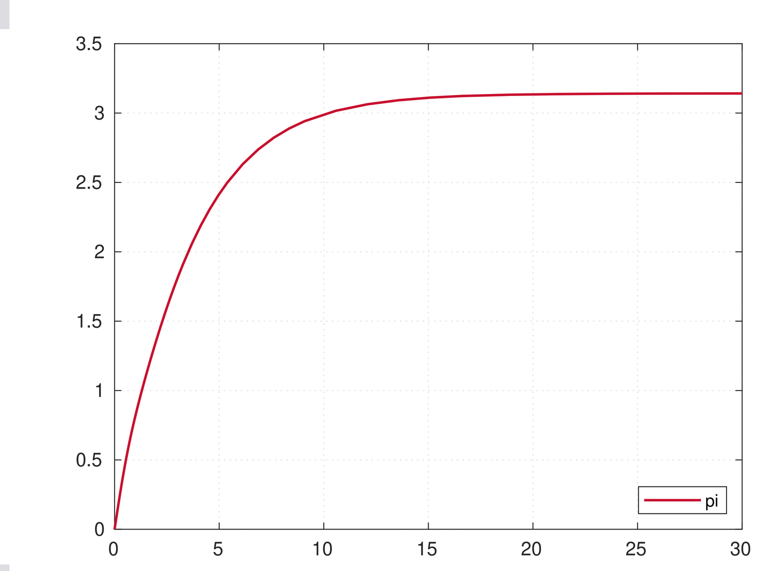
1. Pick $x(t)$ that converges to α **exponentially** fast. (creative)
2. Implement $x(t)$ by a GPAC. (creative)
3. Translate the GPAC into a CRN $\mathcal{N}$. (automatic)
4. Lastly, turn $\mathcal{N}$ into $\widehat{\mathcal{N}}$. (automatic)



Compute π

We pick the function $x(t) = 4\arctan(1 - e^{-t})$. The resulting CRN has a size 10 species and 120 chemical reactions.





* Original tittle of the paper:*Real-Time Equivalence of Chemical Reaction Networks and Analog Computers.*

TikZposter