HiWi Notes: Minimization of the Code Constraint Polynomial using Homotopy Continuation Methods

28.03.2025

Andreas Tsouchlos

Basic Idea of Homotopy Continuation [CL15]

- Goal: Solve system of equations $F(\boldsymbol{x}) = \boldsymbol{0}, \quad F: \mathbb{R}^n \to \mathbb{R}^n$
- Problem: Depending on F, solving this directly may be difficult
- Solution: Define *homotopy function* $H(\boldsymbol{x}, t)$ with

$$H(\boldsymbol{x}, 0) = G(\boldsymbol{x}), \qquad H(\boldsymbol{x}, 1) = F(\boldsymbol{x}),$$

i.e., a deformation between two systems $G(\boldsymbol{x})$ and $F(\boldsymbol{x})$ (where the zeros of G can be easily obtained); E.g.,

$$H(\boldsymbol{x}, t) = (t - 1)G(\boldsymbol{x}) + tF(\boldsymbol{x}).$$

Then, compute $(\boldsymbol{x}_0, 0)$ such that $G(\boldsymbol{x}_0) = \boldsymbol{0}$ and trace path to $(\boldsymbol{x}_1, 1)$ with $F(\boldsymbol{x}_1) = \boldsymbol{0}$

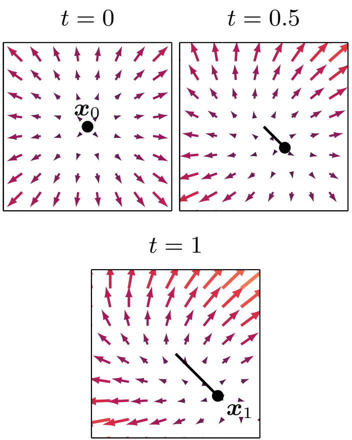


Figure: Visualization of "snapshots" of H (e.g., F, G) as vector fields

[CL15] Chen, Tianran, and Tien-Yien Li.: *Homotopy continuation method for solving systems of nonlinear and polynomial equations*. Communications in Information and Systems 15.2 (2015): 119-307.

Path Tracing

- Reminder: We are trying to trace the solution curve $H(\boldsymbol{x}, t) = \mathbf{0}$ from $t = 0$ to $t = 1$
- We can express the solution curve as a system of differential equations [CL15]:

$$
\begin{aligned}
DH(\boldsymbol{y}(s)) \cdot \dot{\boldsymbol{y}}(s) &= 0 \\
\text{det} \begin{pmatrix} DH(\boldsymbol{y}(s)) \\ \dot{\boldsymbol{y}}(s) \end{pmatrix} &= \sigma_0 \\
\|\dot{\boldsymbol{y}}(s)\| &= 1 \\
\boldsymbol{y}(0) &= (\boldsymbol{x}_0, 0),
\end{aligned}
$$

  where $DH(\boldsymbol{y})$ is the Jacobian of $H(\boldsymbol{y})$ and $\sigma_0 \in \{\pm 1\}$ defines the direction along which we move on the curve.
- For numerical stability, it is beneficial to solve this using a predictor-corrector scheme, e.g., Euler's predictor and Newton's corrector [CL15]:

$$
\begin{aligned}
\hat{\boldsymbol{y}} &= \boldsymbol{y}_0 + \Delta s \cdot \sigma \cdot \boldsymbol{y}(\boldsymbol{s}) \\
\boldsymbol{y} &= \mathcal{N}^k(\hat{\boldsymbol{y}}), \qquad \mathcal{N}(\hat{\boldsymbol{y}}) := \hat{\boldsymbol{y}} - (DH(\hat{\boldsymbol{y}}))^+ H(\hat{\boldsymbol{y}}).
\end{aligned}
$$

[CL15] Chen, Tianran, and Tien-Yien Li.: *Homotopy continuation method for solving systems of nonlinear and polynomial equations.* Communications in Information and Systems 15.2 (2015): 119-307.

Channel Decoding and Polynomial Equations

- To describe the decoding problem we can use the code constraint polynomial [WT22]

$$h(\boldsymbol{x}) = \sum_{i=1}^{n} \left(1 - x_i^2\right)^2 + \sum_{j=1}^{m} \left(1 - \left(\prod_{i \in A(j)} x_i\right)\right)^2 .$$

  where $A(j) = \{i \in [1:n] : \boldsymbol{H}_{j,i} = 1\}, \quad j \in [1:m]$ represents the set of variables involved in parity check j.

- In a similar vein, we can define a polynomial system whose zeros correspond to codewords as

$$F(\boldsymbol{x}) = \begin{bmatrix} 1 - x_1^2 \\ \vdots \\ 1 - x_n^2 \\ 1 - \prod_{i \in A(1)} x_i \\ \vdots \\ 1 - \prod_{i \in A(m)} x_i \end{bmatrix} \stackrel{!}{=} \boldsymbol{0}.$$

[WT22] Tadashi Wadayama; Satoshi Takabe: Proximal Decoding for LDPC Codes. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences advpub (2022), 2022TAP0002.

Defining Homotopies for Channel Codes

- Problem: Homotopy continuation algorithms / existing frameworks only really support square systems, i.e., # equations = # variables. The system $F(\boldsymbol{x}) = \mathbf{0}$ we previously considered is overdefined
- *Gröbner bases* allow us to "[...] transform F into another set G of polynomials [...] such that F and G are equivalent" [B01], i.e., they have the same zeros
- Limited tests indicate that, for the systems we are interested in, finding a Gröbner basis yields a square system
- Example:

$$\overbrace{\boldsymbol{H}}^{\text{Parity check matrix}} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$F(\boldsymbol{x}) = \begin{bmatrix} 1 - x_1^2 \\ 1 - x_2^2 \\ 1 - x_1 x_2 \end{bmatrix} \longrightarrow$$

$$\tilde{F}(\boldsymbol{x}) = \begin{bmatrix} x_1 - x_2 \\ x_2^2 - 1 \end{bmatrix}$$

$$G(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$H(\boldsymbol{x}, t) = (1-t)G(\boldsymbol{x}) + tF(\boldsymbol{x})$$

[B01] Buchberger, Bruno. "Gröbner bases: A short introduction for systems theorists." International Conference on Computer Aided Systems Theory. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

Path Tracker Implementation (Pseudo Code)

- Perform a predictor step followed by multiple corrector steps
- If the corrector fails to converge, adjust the predictor step size and try again [CL15]

```
func perform_prediction_step(y, step_size) {...}
func perform_correction_step(y) {...}
func perform_step(y0) {
    for i in range(max_retries):
        step_size = step_size / 2

        y = perform_prediction_step(y0, step_size)

        for k in range(max_corrector_iterations):
            y = perform_correction_step(y)
            if (corrector converged) break
        if (corrector converged) break

    return y
}
```

[CL15] Chen, Tianran, and Tien-Yien Li.: *Homotopy continuation method for solving systems of nonlinear and polynomial equations*. Communications in Information and Systems 15.2 (2015): 119-307.

Decoding Algorithm Implementation (Pseudo Code)

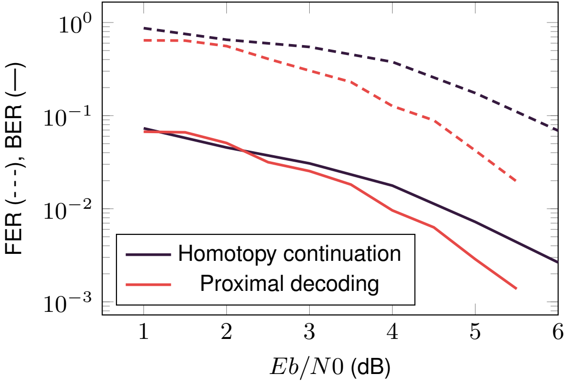
- If the algorithm doesn’t converge, we still return the last estimate in the hopes that this will limit the BER

```
func decode(y) {
    for i in range(max_iterations):
        y = perform_step(y)

        x_hat = hard_decision(y)
        if (H @ x_hat == 0) return x_hat

    return x_hat
}
```

Simulation results

- Simulation using the all-zeros codeword
- Newton homotopy:

$$G(\boldsymbol{x}) = F(x) - F(\boldsymbol{y}) \quad \Rightarrow \quad H(\boldsymbol{x}) = F(\boldsymbol{x}) - (1-t)F(\boldsymbol{y})$$



(a) BCH(31,26) Code

	Parameter	Value
n_{iter}	for homotopy continuation	20
n_{iter}	for Newton corrector	5
δ_{max}	for Newton corrector	0.01
Δs	for Euler predictor	0.05
n_{retries}	for Euler predictor	5

(No comprehensive investigation into choice of parameters completed yet)

Next steps

Karlsruhe Institute of Technology

- Simulations for other codes
- Thorough investigation into parameter choice
- Find more mathematical background / guarantees
  - How do we have to choose σ_0?
  - Guarantees for convergence? (i.e., what is the cause for decoding failures?)
  - When do we actually get square systems using the Gröbner basis?
- Other ideas:
  - Generate more candidates by moving further along the solution curve (if this is possible) and then performing choosing from this list