Jonathan Parlett

February 11, 2023

- Cyclotomic polynomials and their degrees

- Cyclotomic polynomials and their degrees
- A simplified encoding scheme from polynomials with complex coefficients to vectors with complex coefficients

- Cyclotomic polynomials and their degrees
- A simplified encoding scheme from polynomials with complex coefficients to vectors with complex coefficients
- The actual encoding scheme from polynomials with integer coefficients to vectors of complex coefficients.

- The n-th Cyclotomic polynomial is defined as $\Phi_n(x) = \Pi_{1 \leq k \leq n \mid \gcd(k,n)=1}(x - e^{2i\pi\frac{k}{n}})$

- The n-th Cyclotomic polynomial is defined as $\Phi_n(x) = \Pi_{1 \leq k \leq n \,|\, \gcd(k,n)=1}(x - e^{2i\pi\frac{k}{n}})$
- From the constraint that $gcd(k, n) = 1$ you may be able to infer that the degree of the n-th cyclotomic polynomial is equal to $\rho(n)$ where ρ is Eulers totient function.

- The n-th Cyclotomic polynomial is defined as
  $\Phi_n(x) = \Pi_{1 \leq k \leq n \mid \gcd(k,n)=1}(x - e^{2i\pi\frac{k}{n}})$
- From the constraint that $gcd(k, n) = 1$ you may be able to infer that the degree of the n-th cyclotomic polynomial is equal to $\rho(n)$ where ρ is Eulers totient function.
- This property will be important to consider when you we select a cyclotomic to use for our encoding

- The n-th Cyclotomic polynomial is defined as $\Phi_n(x) = \Pi_{1 \leq k \leq n \mid \gcd(k,n)=1}(x - e^{2i\pi\frac{k}{n}})$
- From the constraint that $gcd(k, n) = 1$ you may be able to infer that the degree of the n-th cyclotomic polynomial is equal to $\rho(n)$ where ρ is Eulers totient function.
- This property will be important to consider when you we select a cyclotomic to use for our encoding
- Another important property of cyclotomics is that there roots are complex conjugates of each other. To see this lets look at the 8-th cyclotomic $X^4 + 1$

Cyclotomic polynomials

- The n-th Cyclotomic polynomial is defined as
  $\Phi_n(x) = \Pi_{1 \le k \le n \mid \gcd(k,n)=1}(x - e^{2i\pi\frac{k}{n}})$
- From the constraint that $gcd(k, n) = 1$ you may be able to infer that the degree of the n-th cyclotomic polynomial is equal to $\rho(n)$ where ρ is Eulers totient function.
- This property will be important to consider when you we select a cyclotomic to use for our encoding
- Another important property of cyclotomics is that there roots are complex conjugates of each other. To see this lets look at the 8-th cyclotomic $X^4 + 1$
- $\Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}))(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}))(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2})(x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}))(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2})(x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})$
- Grouping real and imaginary parts we can see that the 1st root is the complex of the 4th and the 2nd root is the complex conjugate of the 3rd

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}))(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2})(x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})$
- Grouping real and imaginary parts we can see that the 1st root is the complex of the 4th and the 2nd root is the complex conjugate of the 3rd
- $= ([x - \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})([x + \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})([x + \frac{\sqrt{2}}{2}] + i\frac{\sqrt{2}}{2})([x - \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})$

Cyclotomic polynomial roots are complex conjugates: Example

- $x^4 + 1 = \Phi_8(x) = (x - e^{2i\pi\frac{1}{8}})(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - e^{2i\pi\frac{7}{8}})$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - e^{2i\pi\frac{3}{8}})(x - e^{2i\pi\frac{5}{8}})(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))(x - (-\frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2}))(x - (\frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2}))$
- $= (x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})(x + \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2})(x - \frac{\sqrt{2}}{2} - i\frac{\sqrt{2}}{2})$
- Grouping real and imaginary parts we can see that the 1st root is the complex of the 4th and the 2nd root is the complex conjugate of the 3rd
- $= ([x - \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})([x + \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})([x + \frac{\sqrt{2}}{2}] + i\frac{\sqrt{2}}{2})([x - \frac{\sqrt{2}}{2}] - i\frac{\sqrt{2}}{2})$
- This fact will become important when we are discussing the mapping of polynomials with integer coefficients to complex vectors

Simplified encoding scheme

- The input messages in the CKKS scheme are vectors of complex numbers $z \in \mathbb{C}^N$ where N is called the degree modulus for reasons that will become obvious

- The input messages in the CKKS scheme are vectors of complex numbers $z \in \mathbb{C}^N$ where N is called the degree modulus for reasons that will become obvious
- All homomorphic operations in the CKKS scheme are performed on polynomials in the ring $\frac{\mathbb{Z}[X]}{X^N+1}$. The homomorphic properties of these operations come as a consequence of the properties of these polynomial rings

Simplified encoding scheme

- The input messages in the CKKS scheme are vectors of complex numbers $z \in \mathbb{C}^N$ where N is called the degree modulus for reasons that will become obvious
- All homomorphic operations in the CKKS scheme are performed on polynomials in the ring $\frac{\mathbb{Z}[X]}{X^N+1}$. The homomorphic properties of these operations come as a consequence of the properties of these polynomial rings
- So the first big thing to understand in order to fully understand CKKS is this encoding scheme. How do we get from our complex vectors to our plaintext polynomials

- The input messages in the CKKS scheme are vectors of complex numbers $z \in \mathbb{C}^N$ where N is called the degree modulus for reasons that will become obvious
- All homomorphic operations in the CKKS scheme are performed on polynomials in the ring $\frac{\mathbb{Z}[X]}{X^N+1}$. The homomorphic properties of these operations come as a consequence of the properties of these polynomial rings
- So the first big thing to understand in order to fully understand CKKS is this encoding scheme. How do we get from our complex vectors to our plaintext polynomials
- The ultimate is goal to be able to fully understand the map that defines the CKKS encoding algorithm

$$\sigma^{-1} : \mathbb{C}^N \to \frac{\mathbb{Z}[X]}{X^N + 1}$$

- The input messages in the CKKS scheme are vectors of complex numbers $z \in \mathbb{C}^N$ where N is called the degree modulus for reasons that will become obvious
- All homomorphic operations in the CKKS scheme are performed on polynomials in the ring $\frac{\mathbb{Z}[X]}{X^N+1}$. The homomorphic properties of these operations come as a consequence of the properties of these polynomial rings
- So the first big thing to understand in order to fully understand CKKS is this encoding scheme. How do we get from our complex vectors to our plaintext polynomials
- The ultimate is goal to be able to fully understand the map that defines the CKKS encoding algorithm

$$\sigma^{-1} : \mathbb{C}^N \to \frac{\mathbb{Z}[X]}{X^N + 1}$$

- How do they achieve this map from complex vectors to polynomials of integer coefficients

- To start we will first understand the simpler map from complex vectors to polynomials with complex coefficients

$$\sigma^{-1} : \mathbb{C}^N \to \frac{\mathbb{C}[X]}{X^N + 1}$$

- To start we will first understand the simpler map from complex vectors to polynomials with complex coefficients

$$\sigma^{-1} : \mathbb{C}^N \to \frac{\mathbb{C}[X]}{X^N + 1}$$

- The forward map may be easier to consider

$$\sigma : \frac{\mathbb{C}[X]}{X^N + 1} \to \mathbb{C}^N$$

- Assume we have our cyclotomic of degree N. Then to map a polynomial of degree N to a vector in $\mathbb{C}^N$ we simply evaluate that polynomial at the N roots our our cyclotomic

- Assume we have our cyclotomic of degree N. Then to map a polynomial of degree N to a vector in $\mathbb{C}^N$ we simply evaluate that polynomial at the N roots our our cyclotomic
- Our output vector $z \in \mathbb{C}^N$ will be the vector of b_is for all roots of our cyclotomic which is ultimately the result of this matrix vector product for $\Phi_8(x) = X^4 + 1$

- Assume we have our cyclotomic of degree N. Then to map a polynomial of degree N to a vector in $\mathbb{C}^N$ we simply evaluate that polynomial at the N roots our our cyclotomic
- Our output vector $z \in \mathbb{C}^N$ will be the vector of b_is for all roots of our cyclotomic which is ultimately the result of this matrix vector product for $\Phi_8(x) = X^4 + 1$
  -

$$\begin{pmatrix} 1 & (e^i\pi/4)^1 & (e^i\pi/4)^2 & (e^i\pi/4)^3 & (e^i\pi/4)^4 \\ 1 & (e^i\pi 3/4)^1 & (e^i\pi 3/4)^2 & (e^i\pi 3/4)^3 & (e^i\pi 3/4)^4 \\ 1 & (e^i\pi 5/4)^1 & (e^i\pi 5/4)^2 & (e^i\pi 5/4)^3 & (e^i\pi 5/4)^4 \\ 1 & (e^i\pi 7/4)^1 & (e^i\pi 7/4)^2 & (e^i\pi 7/4)^3 & (e^i\pi 7/4)^4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

Simplified encoding scheme

- $$\begin{pmatrix} 1 & (e^i\pi/4)^1 & (e^i\pi/4)^2 & (e^i\pi/4)^3 & (e^i\pi/4)^4 \\ 1 & (e^i\pi 3/4)^1 & (e^i\pi 3/4)^2 & (e^i\pi 3/4)^3 & (e^i\pi 3/4)^4 \\ 1 & (e^i\pi 5/4)^1 & (e^i\pi 5/4)^2 & (e^i\pi 5/4)^3 & (e^i\pi 5/4)^4 \\ 1 & (e^i\pi 7/4)^1 & (e^i\pi 7/4)^2 & (e^i\pi 7/4)^3 & (e^i\pi 7/4)^4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

■

$$\begin{pmatrix} 1 & (e^i\pi/4)^1 & (e^i\pi/4)^2 & (e^i\pi/4)^3 & (e^i\pi/4)^4 \\ 1 & (e^i\pi3/4)^1 & (e^i\pi3/4)^2 & (e^i\pi3/4)^3 & (e^i\pi3/4)^4 \\ 1 & (e^i\pi5/4)^1 & (e^i\pi5/4)^2 & (e^i\pi5/4)^3 & (e^i\pi5/4)^4 \\ 1 & (e^i\pi7/4)^1 & (e^i\pi7/4)^2 & (e^i\pi7/4)^3 & (e^i\pi7/4)^4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

- Here we can see we have the coefficient vector **a** that uniquely determines the polynomial and the output vector **z** $\in \mathbb{C}^N$.

■

$$\begin{pmatrix} 1 & (e^i\pi/4)^1 & (e^i\pi/4)^2 & (e^i\pi/4)^3 & (e^i\pi/4)^4 \\ 1 & (e^i\pi3/4)^1 & (e^i\pi3/4)^2 & (e^i\pi3/4)^3 & (e^i\pi3/4)^4 \\ 1 & (e^i\pi5/4)^1 & (e^i\pi5/4)^2 & (e^i\pi5/4)^3 & (e^i\pi5/4)^4 \\ 1 & (e^i\pi7/4)^1 & (e^i\pi7/4)^2 & (e^i\pi7/4)^3 & (e^i\pi7/4)^4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

- Here we can see we have the coefficient vector **a** that uniquely determines the polynomial and the output vector **z** $\in \mathbb{C}^N$.
- Its clear from this equation that given one we can compute the other and since this is a square matrix there is one and only one solution

- $$\begin{pmatrix} 1 & (e^i\pi/4)^1 & (e^i\pi/4)^2 & (e^i\pi/4)^3 & (e^i\pi/4)^4 \\ 1 & (e^i\pi 3/4)^1 & (e^i\pi 3/4)^2 & (e^i\pi 3/4)^3 & (e^i\pi 3/4)^4 \\ 1 & (e^i\pi 5/4)^1 & (e^i\pi 5/4)^2 & (e^i\pi 5/4)^3 & (e^i\pi 5/4)^4 \\ 1 & (e^i\pi 7/4)^1 & (e^i\pi 7/4)^2 & (e^i\pi 7/4)^3 & (e^i\pi 7/4)^4 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix} = \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}$$

- Here we can see we have the coefficient vector **a** that uniquely determines the polynomial and the output vector **z** $\in \mathbb{C}^N$.
- Its clear from this equation that given one we can compute the other and since this is a square matrix there is one and only one solution
- so it should be intuitive that this transformation defines an isomorphism between $\mathbb{C}^4$ and $\frac{\mathbb{C}[X]}{X^4+1}$

- At this point we have defined our simplified map and its inverse as essentially the equation we showed in the previous slide

$$\sigma^{-1} : \mathbb{C}^N \to \frac{\mathbb{C}[X]}{X^N + 1}$$

- At this point we have defined our simplified map and its inverse as essentially the equation we showed in the previous slide
$$\sigma^{-1} : \mathbb{C}^N \rightarrow \frac{\mathbb{C}[X]}{X^N + 1}$$
- The CKKS map/algorithm $\sigma^{-1} : \mathbb{C}^N \rightarrow \frac{\mathbb{Z}[X]}{X^N+1}$ adds further structure in order to place restrictions on this map to ensure that we encode our complex vectors as polynomials with integer coefficients only

- To understand something about the structure of our map well need the fact that evaluating polynomials with real coefficients at complex conjugates produces complex conjugates

- To understand something about the structure of our map well need the fact that evaluating polynomials with real coefficients at complex conjugates produces complex conjugates
- This should be straight forward if we understand the simpler statement that powers of complex conjugates are still complex conjugates

- To understand something about the structure of our map well need the fact that evaluating polynomials with real coefficients at complex conjugates produces complex conjugates
- This should be straight forward if we understand the simpler statement that powers of complex conjugates are still complex conjugates
- This is simple to show using Euler's formula so we will state it here for clarity

- For some $z \in \mathbb{C}$ we have that $z = re^{ix}$ for some $r, x \in \mathbb{R}$

Evaluating polynomials at complex coefficients

- For some $z \in \mathbb{C}$ we have that $z = re^{ix}$ for some $r, x \in \mathbb{R}$
- Then its conjugate $\overline{z} = re^{-ix}$

- For some $z \in \mathbb{C}$ we have that $z = re^{ix}$ for some $r, x \in \mathbb{R}$
- Then its conjugate $\overline{z} = re^{-ix}$
-

$$(\overline{z})^n = (re^{-ix})^n \tag{1}$$
$$(\overline{z})^n = (re^{-nix}) \tag{2}$$
$$(\overline{z})^n = (re^{nix})^{-1} \tag{3}$$
$$(\overline{z})^n = (z^n)^{-1} \tag{4}$$
$$(\overline{z})^n = \overline{z^n} \tag{5}$$
$$\tag{6}$$

- For some $z \in \mathbb{C}$ we have that $z = re^{ix}$ for some $r, x \in \mathbb{R}$
- Then its conjugate $\overline{z} = re^{-ix}$
-

$$(\overline{z})^n = (re^{-ix})^n \tag{1}$$
$$(\overline{z})^n = (re^{-nix}) \tag{2}$$
$$(\overline{z})^n = (re^{nix})^{-1} \tag{3}$$
$$(\overline{z})^n = (z^n)^{-1} \tag{4}$$
$$(\overline{z})^n = \overline{z^n} \tag{5}$$
$$\tag{6}$$

- Then if we constructed a general polynomial with real coefficients we could use this fact to make our conclusion

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$
- Then for the conjugate we have

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$
- Then for the conjugate we have
- $P(2 + i) = (2 + i)^2 + 1$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$
- Then for the conjugate we have
- $P(2 + i) = (2 + i)^2 + 1$
- $P(2 + i) = 4 + 4i + i^2 + 1$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$
- Then for the conjugate we have
- $P(2 + i) = (2 + i)^2 + 1$
- $P(2 + i) = 4 + 4i + i^2 + 1$
- $P(2 + i) = 4 + 4i$

Evaluating polynomials at complex coefficients:Exampe

- For complex conjugates $2 - i, 2 + i$ and polynomial $P(x) = x^2 + 1$
- $P(2 - i) = (2 - i)^2 + 1$
- $P(2 - i) = 4 - 4i + i^2 + 1$
- $P(2 - i) = 4 - 4i$
- Then for the conjugate we have
- $P(2 + i) = (2 + i)^2 + 1$
- $P(2 + i) = 4 + 4i + i^2 + 1$
- $P(2 + i) = 4 + 4i$
- So we can see that they are conjugates

- Does this hold for polynomials with complex coefficients?

Evaluating polynomials at complex coefficients

- Does this hold for polynomials with complex coefficients?
- No to see why just consider that multiplying 2 complex conjugates by another complex number does not necessarily produce complex conjugates

- Does this hold for polynomials with complex coefficients?
- No to see why just consider that multiplying 2 complex conjugates by another complex number does not necessarily produce complex conjugates
- Now that we have shown this we can always say for any complex z and polynomial with real coefficients $P(x)$, $\overline{P(z)} = P(\overline{z})$

- Does this hold for polynomials with complex coefficients?
- No to see why just consider that multiplying 2 complex conjugates by another complex number does not necessarily produce complex conjugates
- Now that we have shown this we can always say for any complex z and polynomial with real coefficients $P(x)$, $\overline{P(z)} = P(\overline{z})$
- Now lets return to CKKS and our map $\sigma : \frac{\mathbb{C}[X]}{X^N+1} \to \mathbb{C}^N$

Integer Polynomials map to vectors of complex conjugates

- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only

Integer Polynomials map to vectors of complex conjugates

- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only
- So a good place to start would be looking at what integer coefficient polynomials map to under our current transformation σ^{-1}

Integer Polynomials map to vectors of complex conjugates

- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only
- So a good place to start would be looking at what integer coefficient polynomials map to under our current transformation σ^{-1}
- Lets first define $R = \frac{\mathbb{Z}[X]}{X^N+1}$ to be the space of polynomials with integer coefficients

Integer Polynomials map to vectors of complex conjugates

- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only
- So a good place to start would be looking at what integer coefficient polynomials map to under our current transformation σ^{-1}
- Lets first define $R = \frac{\mathbb{Z}[X]}{X^N+1}$ to be the space of polynomials with integer coefficients
- So more formally our goal is to define the image of R under our transformation σ denoted $\sigma(R)$

Integer Polynomials map to vectors of complex conjugates

- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only
- So a good place to start would be looking at what integer coefficient polynomials map to under our current transformation σ^{-1}
- Lets first define $R = \frac{\mathbb{Z}[X]}{X^N+1}$ to be the space of polynomials with integer coefficients
- So more formally our goal is to define the image of R under our transformation σ denoted $\sigma(R)$
- To map some $P(x) \in R$ to $\mathbb{C}^N$ we evaluate at the roots of our cyclotomic

Integer Polynomials map to vectors of complex conjugates

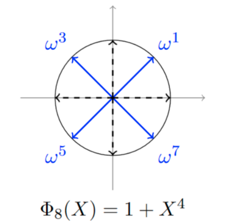
- We eventually want to figure out how to map our complex vectors to polynomials with integer coefficients only
- So a good place to start would be looking at what integer coefficient polynomials map to under our current transformation σ^{-1}
- Lets first define $R = \frac{\mathbb{Z}[X]}{X^N+1}$ to be the space of polynomials with integer coefficients
- So more formally our goal is to define the image of R under our transformation σ denoted $\sigma(R)$
- To map some $P(x) \in R$ to $\mathbb{C}^N$ we evaluate at the roots of our cyclotomic
- Recall that the roots of our cyclotomic are complex conjugates of each other.

Integer Polynomials map to vectors of complex conjugates

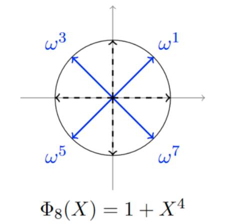
- So evaluating our polynomial at the roots of our cyclotomic produces a vector of complex conjugates because of the property we showed previously

Integer Polynomials map to vectors of complex conjugates

- So evaluating our polynomial at the roots of our cyclotomic produces a vector of complex conjugates because of the property we showed previously
- For a more concrete picture imagine we have the 4 roots of the 8th cyclotomic $\Phi_8(x) = X^4 + 1$

- So evaluating our polynomial at the roots of our cyclotomic produces a vector of complex conjugates because of the property we showed previously
- For a more concrete picture imagine we have the 4 roots of the 8th cyclotomic $\Phi_8(x) = X^4 + 1$
- 

Integer Polynomials map to vectors of complex conjugates

- So evaluating our polynomial at the roots of our cyclotomic produces a vector of complex conjugates because of the property we showed previously
- For a more concrete picture imagine we have the 4 roots of the 8th cyclotomic $\Phi_8(x) = X^4 + 1$
- 

- We can see that $\omega^1 = \overline{\omega^7}$ and $\omega^3 = \overline{\omega^5}$

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- Then this vector is precisely the vector output by our map σ

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- Then this vector is precisely the vector output by our map σ
- $\sigma(P(x)) = \langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- Then this vector is precisely the vector output by our map σ
- $\sigma(P(x)) = \langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- And since $\overline{P(z)} = P(\overline{z})$ for any complex z we have that $P(\omega^1) = \overline{P(\omega^7)}$, and so on

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- Then this vector is precisely the vector output by our map σ
- $\sigma(P(x)) = \langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- And since $\overline{P(z)} = P(\overline{z})$ for any complex z we have that $P(\omega^1) = \overline{P(\omega^7)}$, and so on
- So we have that any polynomial with real coefficients maps to a vector of complex conjugates

Integer Polynomials map to vectors of complex conjugates

- Take some polynomial $P(x)$ with real coefficients and use it to transform the vector $\vec{z} = \langle \omega^1, \omega^3, \omega^5, \omega^7 \rangle$, we can see this satisfies the property $z_i = \overline{z_{4-i}}$
- $\langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- Then this vector is precisely the vector output by our map σ
- $\sigma(P(x)) = \langle P(\omega^1), P(\omega^3), P(\omega^5), P(\omega^7) \rangle$
- And since $\overline{P(z)} = P(\overline{z})$ for any complex z we have that $P(\omega^1) = \overline{P(\omega^7)}$, and so on
- So we have that any polynomial with real coefficients maps to a vector of complex conjugates
- Based on how we chose to order the conjugates it specifically maps to a complex vector $\vec{z} \in \mathbb{C}^N$ s.t $z_i = \overline{z_{N-i}}$

The image of R

- Now consider the set of all complex vectors who's i-th coordinates are the complex conjugate of their $N - i$-th

$$\mathbb{H} = \{z \in \mathbb{C}^N | z_i = \overline{z_{N-i}}\} \subset \mathbb{C}^N$$

The image of R

- Now consider the set of all complex vectors who's i-th coordinates are the complex conjugate of their $N - i$-th

$$\mathbb{H} = \{z \in \mathbb{C}^N | z_i = \overline{z_{N-i}}\} \subset \mathbb{C}^N$$

- Now any polynomial with real coefficients maps to a vector in $\mathbb{H}$

- Now consider the set of all complex vectors who's i-th coordinates are the complex conjugate of their $N - i$-th

$$\mathbb{H} = \{z \in \mathbb{C}^N | z_i = \overline{z_{N-i}}\} \subset \mathbb{C}^N$$

- Now any polynomial with real coefficients maps to a vector in $\mathbb{H}$
- So necessarily any polynomial with integer coefficients maps to a vector in $\mathbb{H}$

- Now consider the set of all complex vectors who's i-th coordinates are the complex conjugate of their $N-i$-th

$$\mathbb{H} = \{z \in \mathbb{C}^N | z_i = \overline{z_{N-i}}\} \subset \mathbb{C}^N$$

- Now any polynomial with real coefficients maps to a vector in $\mathbb{H}$
- So necessarily any polynomial with integer coefficients maps to a vector in $\mathbb{H}$
- Now stepping back we can see that we have reached our goal of defining the image of R. Namely it is a subset of $\mathbb{H}$

$$\sigma(R) \subset \mathbb{H} = \{z \in \mathbb{C}^N | z_i = \overline{z_{N-i}}\}$$

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$
- A first step towards this might be restricting our input vectors to only those complex vectors in $\mathbb{H}$

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$
- A first step towards this might be restricting our input vectors to only those complex vectors in $\mathbb{H}$
- But how can we do this while preserving the generality of our inputs? We don't want to not be able to encode certain messages IE certain vectors in $\mathbb{C}^N$

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$
- A first step towards this might be restricting our input vectors to only those complex vectors in $\mathbb{H}$
- But how can we do this while preserving the generality of our inputs? We don't want to not be able to encode certain messages IE certain vectors in $\mathbb{C}^N$
- CKKS solves this problem by instead considering the input space as $\mathbb{C}^{N/2}$ and defining the map

$$\pi^{-1} : \mathbb{C}^{N/2} \rightarrow \mathbb{H}$$

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$
- A first step towards this might be restricting our input vectors to only those complex vectors in $\mathbb{H}$
- But how can we do this while preserving the generality of our inputs? We don't want to not be able to encode certain messages IE certain vectors in $\mathbb{C}^N$
- CKKS solves this problem by instead considering the input space as $\mathbb{C}^{N/2}$ and defining the map

$$\pi^{-1} : \mathbb{C}^{N/2} \to \mathbb{H}$$

- The map itself is rather simplistic. It takes a vector in $\mathbb{C}^{N/2}$ and doubles its size by copying all coordinates and conjugating them s.t we have a vector in $\mathbb{H} \subset \mathbb{C}^N$

Changing the input space

- Now ultimately we want to be able to map any complex vector to $\sigma(R)$
- A first step towards this might be restricting our input vectors to only those complex vectors in $\mathbb{H}$
- But how can we do this while preserving the generality of our inputs? We don't want to not be able to encode certain messages IE certain vectors in $\mathbb{C}^N$
- CKKS solves this problem by instead considering the input space as $\mathbb{C}^{N/2}$ and defining the map

$$\pi^{-1} : \mathbb{C}^{N/2} \to \mathbb{H}$$

- The map itself is rather simplistic. It takes a vector in $\mathbb{C}^{N/2}$ and doubles its size by copying all coordinates and conjugating them s.t we have a vector in $\mathbb{H} \subset \mathbb{C}^N$
- The inverse map $\pi(z) \in \mathbb{C}^{N/2}$ simply cuts the vector in half discarding the 2nd half of conjugates

- Now by composing these maps we can get from any complex vector to a polynomial with real coefficients

$$(\sigma^{-1} \circ \pi^{-1})(z) : \mathbb{C}^{N/2} \to \frac{\mathbb{R}[x]}{X^N + 1}$$

- Now by composing these maps we can get from any complex vector to a polynomial with real coefficients

$$(\sigma^{-1} \circ \pi^{-1})(z) : \mathbb{C}^{N/2} \rightarrow \frac{\mathbb{R}[x]}{X^N + 1}$$

- So almost there, now we just need to narrow our map to get to integer polynomials only. The next step is to define a map from $\mathbb{H} \rightarrow \sigma(R)$

Changing the input space

- Now by composing these maps we can get from any complex vector to a polynomial with real coefficients

$$(\sigma^{-1} \circ \pi^{-1})(z) : \mathbb{C}^{N/2} \to \frac{\mathbb{R}[x]}{X^N + 1}$$

- So almost there, now we just need to narrow our map to get to integer polynomials only. The next step is to define a map from $\mathbb{H} \to \sigma(R)$
- This process in the paper is described as the discretization of $\pi^{-1}(z)$ to $\sigma(R)$

- Now by composing these maps we can get from any complex vector to a polynomial with real coefficients

$$(\sigma^{-1} \circ \pi^{-1})(z) : \mathbb{C}^{N/2} \rightarrow \frac{\mathbb{R}[x]}{X^N + 1}$$

- So almost there, now we just need to narrow our map to get to integer polynomials only. The next step is to define a map from $\mathbb{H} \rightarrow \sigma(R)$
- This process in the paper is described as the discretization of $\pi^{-1}(z)$ to $\sigma(R)$
- This is however where I get pretty lost but I will present the theory covered in the paper regardless

Discretization to $\sigma(R)$

- Now R has a $\mathbb{Z}$-basis $\{1, X, \ldots, X^{N-1}\}$

- Now R has a $\mathbb{Z}$-basis $\{1, X, ..., X^{N-1}\}$
- This is saying that any polynomial in R can be expressed as a linear combination of the polynomials in this $\mathbb{Z}$-basis

- Now R has a $\mathbb{Z}$-basis $\{1, X, ..., X^{N-1}\}$
- This is saying that any polynomial in R can be expressed as a linear combination of the polynomials in this $\mathbb{Z}$-basis
- Then this basis maps to a rank N ideal lattice $\sigma(R)$ having basis $\{\sigma(1), \sigma(X), ..., \sigma(X^{N-1})\}$

Discretization to $\sigma(R)$

- Now R has a $\mathbb{Z}$-basis $\{1, X, \ldots, X^{N-1}\}$
- This is saying that any polynomial in R can be expressed as a linear combination of the polynomials in this $\mathbb{Z}$-basis
- Then this basis maps to a rank N ideal lattice $\sigma(R)$ having basis $\{\sigma(1), \sigma(X), \ldots, \sigma(X^{N-1})\}$
- This means we have a set of basis vectors for $\sigma(R)$, that constitute a lattice

- Now R has a $\mathbb{Z}$-basis $\{1, X, ..., X^{N-1}\}$
- This is saying that any polynomial in R can be expressed as a linear combination of the polynomials in this $\mathbb{Z}$-basis
- Then this basis maps to a rank N ideal lattice $\sigma(R)$ having basis $\{\sigma(1), \sigma(X), ..., \sigma(X^{N-1})\}$
- This means we have a set of basis vectors for $\sigma(R)$, that constitute a lattice
- From my understanding the goal is essentially to compute the closest lattice vector to our given input vector thereby transforming our input into a vector that maps to a polynomial in R

Discretization to $\sigma(R)$

- In the paper they do this by first projecting the input vector to the lattice basis, and then via a coordinate wise random rounding algorithm fully discretize to a vector in $\sigma(R)$

Discretization to $\sigma(R)$

- In the paper they do this by first projecting the input vector to the lattice basis, and then via a coordinate wise random rounding algorithm fully discretize to a vector in $\sigma(R)$
- This operation/map is denoted $\lfloor \cdot \rceil_{\sigma(R)}$

Discretization to $\sigma(R)$

- In the paper they do this by first projecting the input vector to the lattice basis, and then via a coordinate wise random rounding algorithm fully discretize to a vector in $\sigma(R)$
- This operation/map is denoted $\lfloor \cdot \rceil_{\sigma(R)}$
- Then we now have a way to get $\mathbb{C}^{N/2} \to \sigma(R)$ via a composition of maps

$$\sigma^{-1} \circ \lfloor \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} : \mathbb{C}^{N/2} \to \sigma(R)$$

Scaling to preserve precision

- They also note that the error resulting from rounding may destroy significant figures of the message so they recommend multiplying by a scaling factor Δ before rounding to preserve precision

- They also note that the error resulting from rounding may destroy significant figures of the message so they recommend multiplying by a scaling factor Δ before rounding to preserve precision
- This changes our current map to

$$\sigma^{-1} \circ \lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} : \mathbb{C}^{N/2} \to \sigma(R)$$

- Now we can state the CKKS encoding algorithm in full

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$
- Expand it to $\pi^{-1}(z) \in \mathbb{H}$

The completed encoding algorithm

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$
- Expand it to $\pi^{-1}(z) \in \mathbb{H}$
- Multiply by Δ to preserve the desired level of precision

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$
- Expand it to $\pi^{-1}(z) \in \mathbb{H}$
- Multiply by Δ to preserve the desired level of precision
- Project to onto our ideal lattice basis via coordinate wise random rounding $\lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} \in \sigma(R)$

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$
- Expand it to $\pi^{-1}(z) \in \mathbb{H}$
- Multiply by Δ to preserve the desired level of precision
- Project to onto our ideal lattice basis via coordinate wise random rounding $\lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} \in \sigma(R)$
- Finally encode it as a polynomial using σ^{-1}, $\sigma^{-1} \circ \lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} \in R$

The completed encoding algorithm

- Now we can state the CKKS encoding algorithm in full
- Take an element $z \in \mathbb{C}^{N/2}$
- Expand it to $\pi^{-1}(z) \in \mathbb{H}$
- Multiply by Δ to preserve the desired level of precision
- Project to onto our ideal lattice basis via coordinate wise random rounding $\lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} \in \sigma(R)$
- Finally encode it as a polynomial using σ^{-1}, $\sigma^{-1} \circ \lfloor \Delta \cdot \pi^{-1}(\mathbf{z}) \rceil_{\sigma(R)} \in R$
- The decoding procedure is just the inverse of the encoding procedure. We simply apply the inverse maps in reverse order to recover the encoded plaintext

- That completes our coverage of the encoding and decoding in CKKS. The entire process is summarized succinctly in this graphic

The completed encoding algorithm

- That completes our coverage of the encoding and decoding in CKKS. The entire process is summarized succinctly in this graphic
- $$\begin{array}{ccccccc}
\mathbb{C}^{\phi(M)/2} & \xrightarrow{\pi^{-1}} & \mathbb{H} & \xrightarrow{\lfloor\cdot\rceil_{\sigma(\mathcal{R})}} & \sigma(\mathcal{R}) & \xrightarrow{\sigma^{-1}} & \mathcal{R} \\
\boldsymbol{z} = (z_i)_{i\in T} & \longmapsto & \pi^{-1}(\boldsymbol{z}) & \longmapsto & \left\lfloor \pi^{-1}(\boldsymbol{z}) \right\rceil_{\sigma(\mathcal{R})} & \longmapsto & \sigma^{-1}\left(\left\lfloor \pi^{-1}(\boldsymbol{z}) \right\rceil_{\sigma(\mathcal{R})} \right)
\end{array}$$

The completed encoding algorithm

- That completes our coverage of the encoding and decoding in CKKS. The entire process is summarized succinctly in this graphic
- $$\begin{array}{ccccccc} \mathbb{C}^{\phi(M)/2} & \xrightarrow{\pi^{-1}} & \mathbb{H} & \xrightarrow{\lfloor \cdot \rceil_{\sigma(\mathcal{R})}} & \sigma(\mathcal{R}) & \xrightarrow{\sigma^{-1}} & \mathcal{R} \\ \boldsymbol{z} = (z_i)_{i \in T} & \longmapsto & \pi^{-1}(\boldsymbol{z}) & \longmapsto & \left\lfloor \pi^{-1}(\boldsymbol{z}) \right\rceil_{\sigma(\mathcal{R})} & \longmapsto & \sigma^{-1}\left(\left\lfloor \pi^{-1}(\boldsymbol{z}) \right\rceil_{\sigma(\mathcal{R})} \right) \end{array}$$
- Thank you for your time!