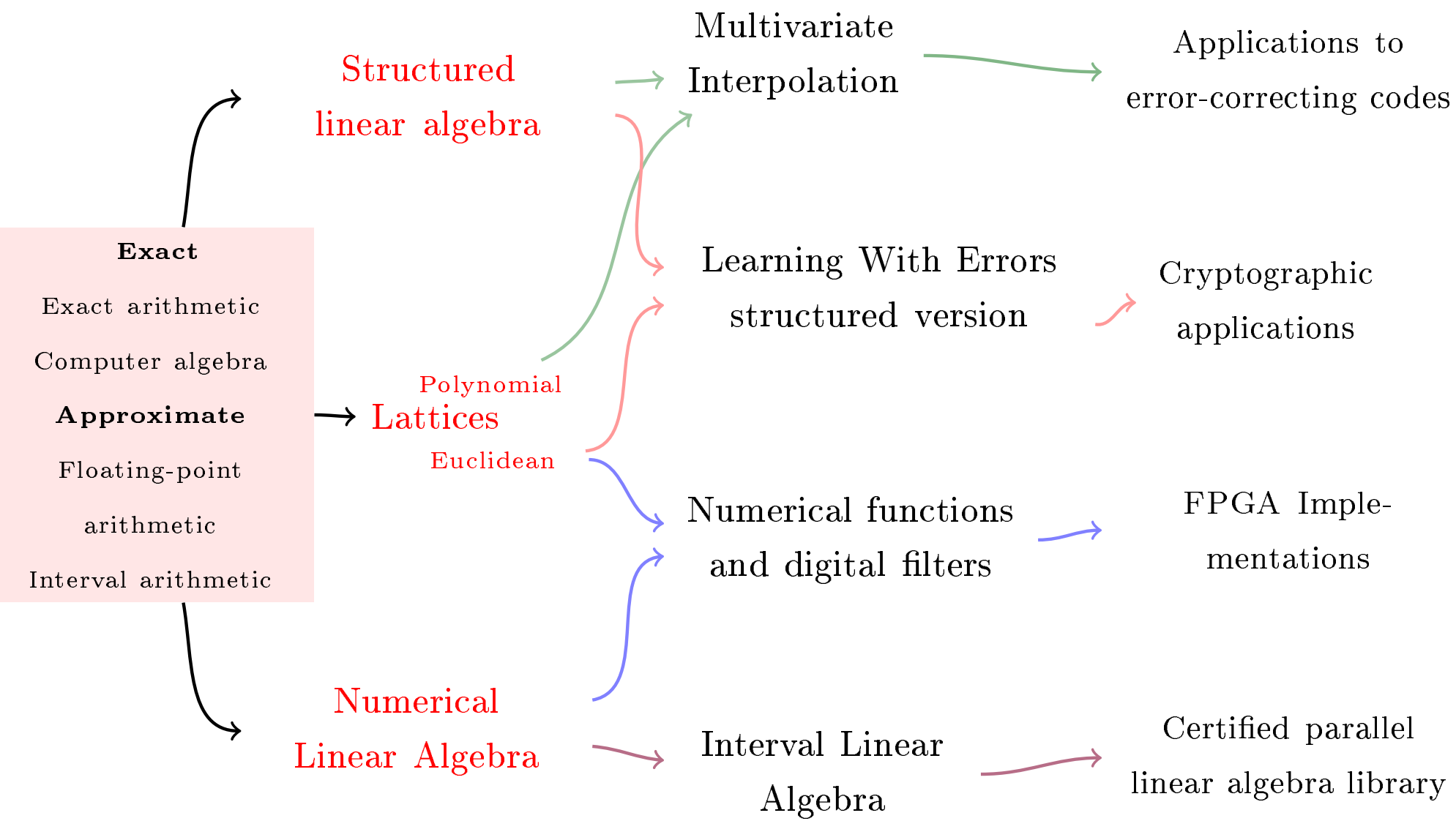
Journée des doctorants

Silviu Filip Sébastien Maulat Stephen Melczer
Vincent Neiger Marie Paindavoine
Antoine Plet Valentina Popescu

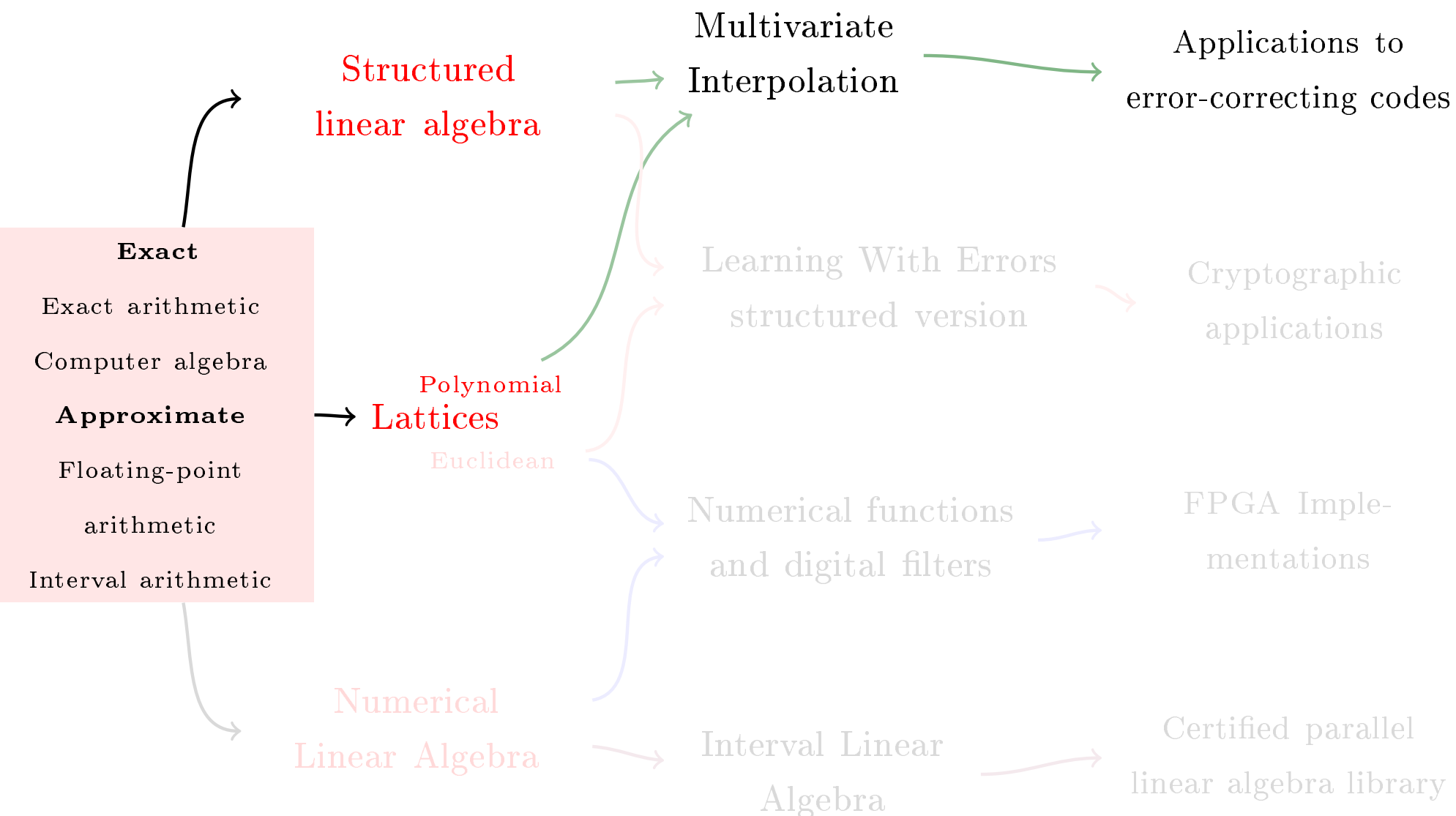
Aric Team, LIP, ENS de Lyon, France

June 2015

AriC


Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point
arithmetic
Interval arithmetic
Structured
linear algebra
Polynomial
Lattices
Euclidean
Numerical
Linear Algebra
Multivariate
Interpolation
Learning With Errors
structured version
Numerical functions
and digital filters
Interval Linear
Algebra
Applications to
error-correcting codes
Cryptographic
applications
FPGA Imple-
mentations
Certified parallel
linear algebra library

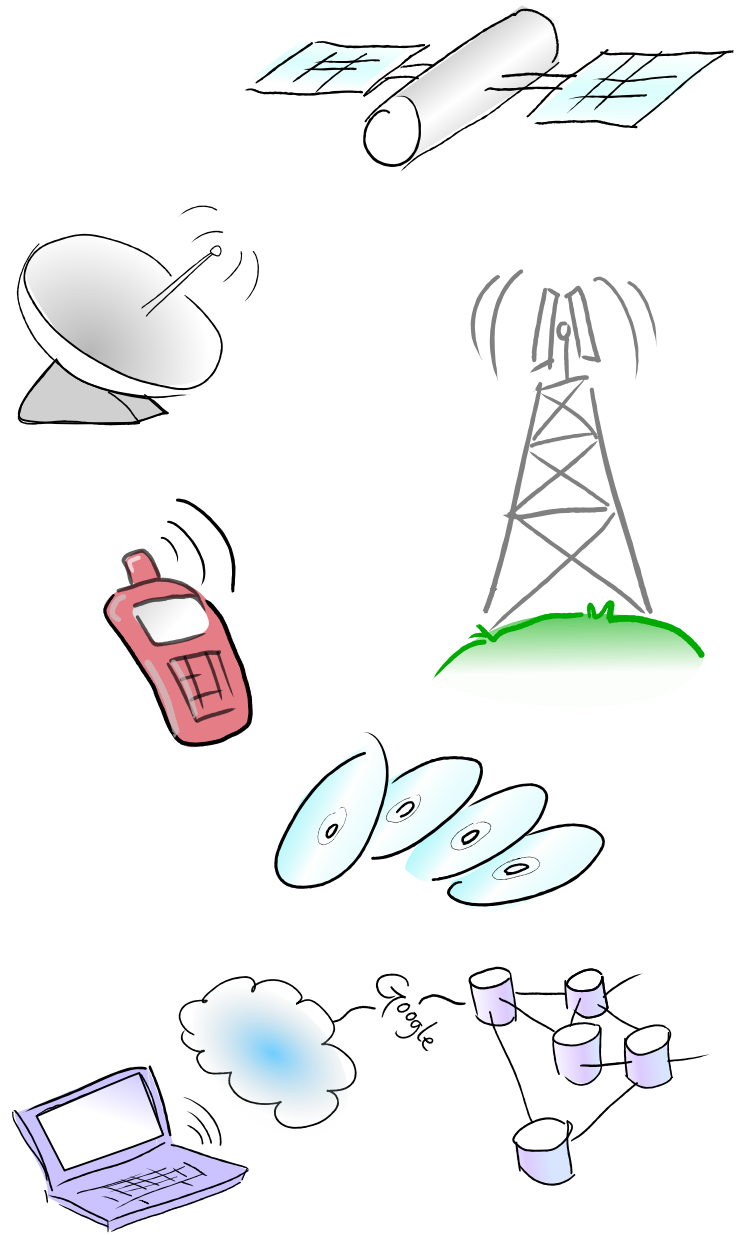
AriC


Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point
arithmetic
Interval arithmetic
Structured
linear algebra
Polynomial
Lattices
Euclidean
Numerical
Linear Algebra
Multivariate
Interpolation
Learning With Errors
structured version
Numerical functions
and digital filters
Interval Linear
Algebra
Applications to
error-correcting codes
Cryptographic
applications
FPGA Imple-
mentations
Certified parallel
linear algebra library

AriC — Error-correcting codes

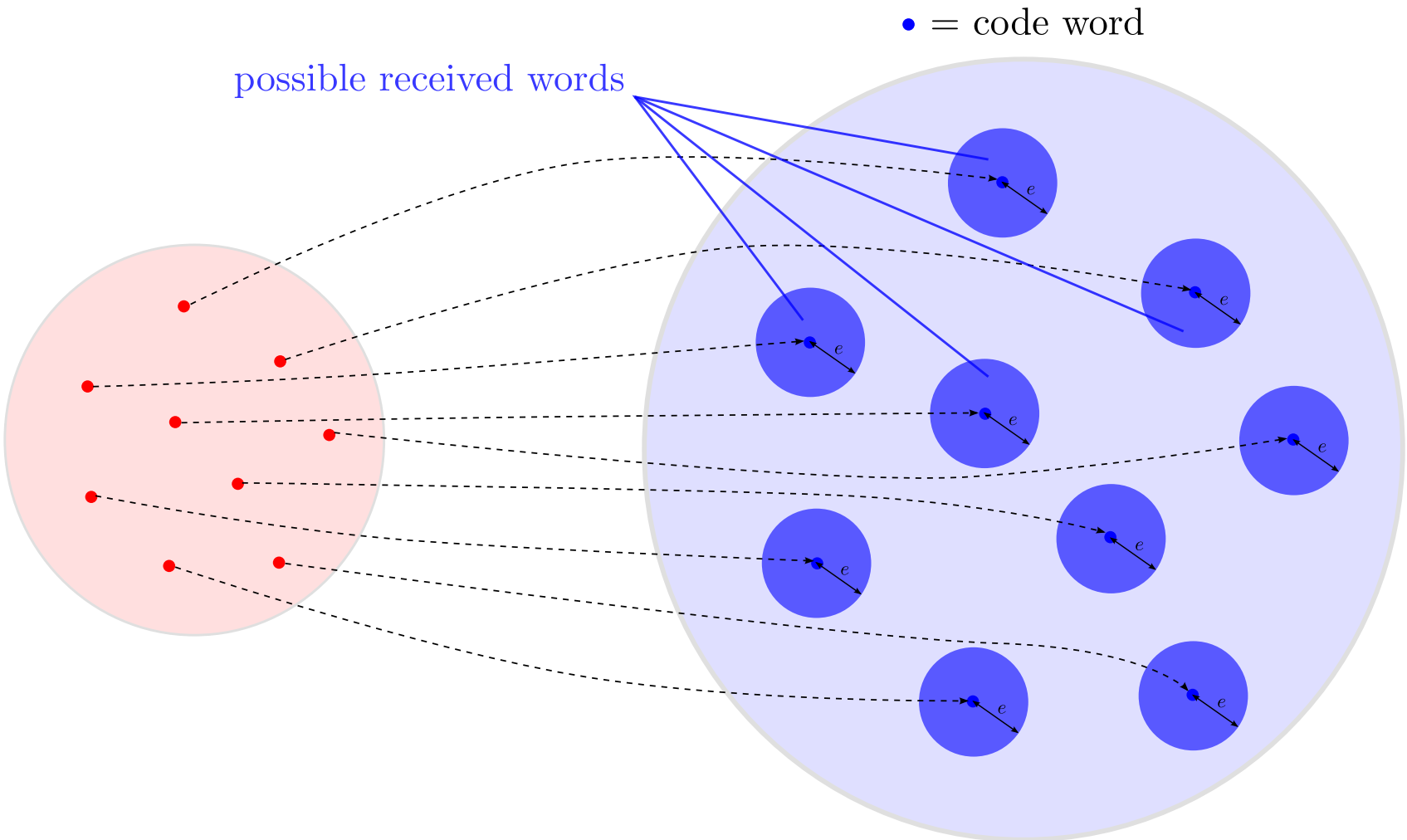
Goal:
Enable reliable delivery of data over unreliable communication channels

Strategy:
add redundancy to the message
add redundancy to the message
add redundancy to the message



(courtesy of J.S.R. Nielsen)

AriC — Error-correcting codes

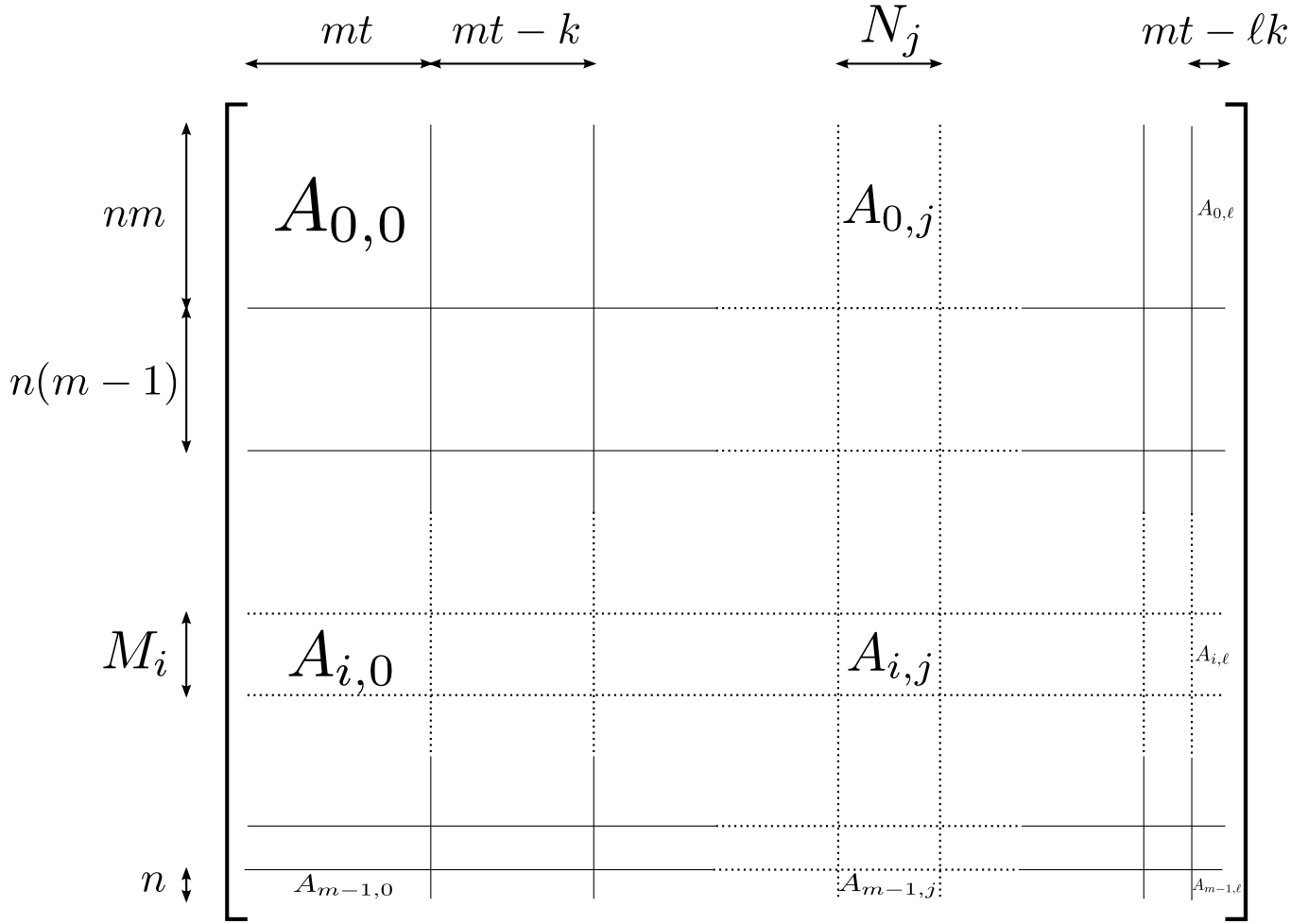


polynomials of degree $\leqslant k$ $\longrightarrow$ their evaluation at $x_1, \ldots, x_n$

$w = w_0 + w_1 X + \cdots + w_k X^k$ $(w(x_1), \ldots, w(x_n))$

AriC — (list-)Decoding

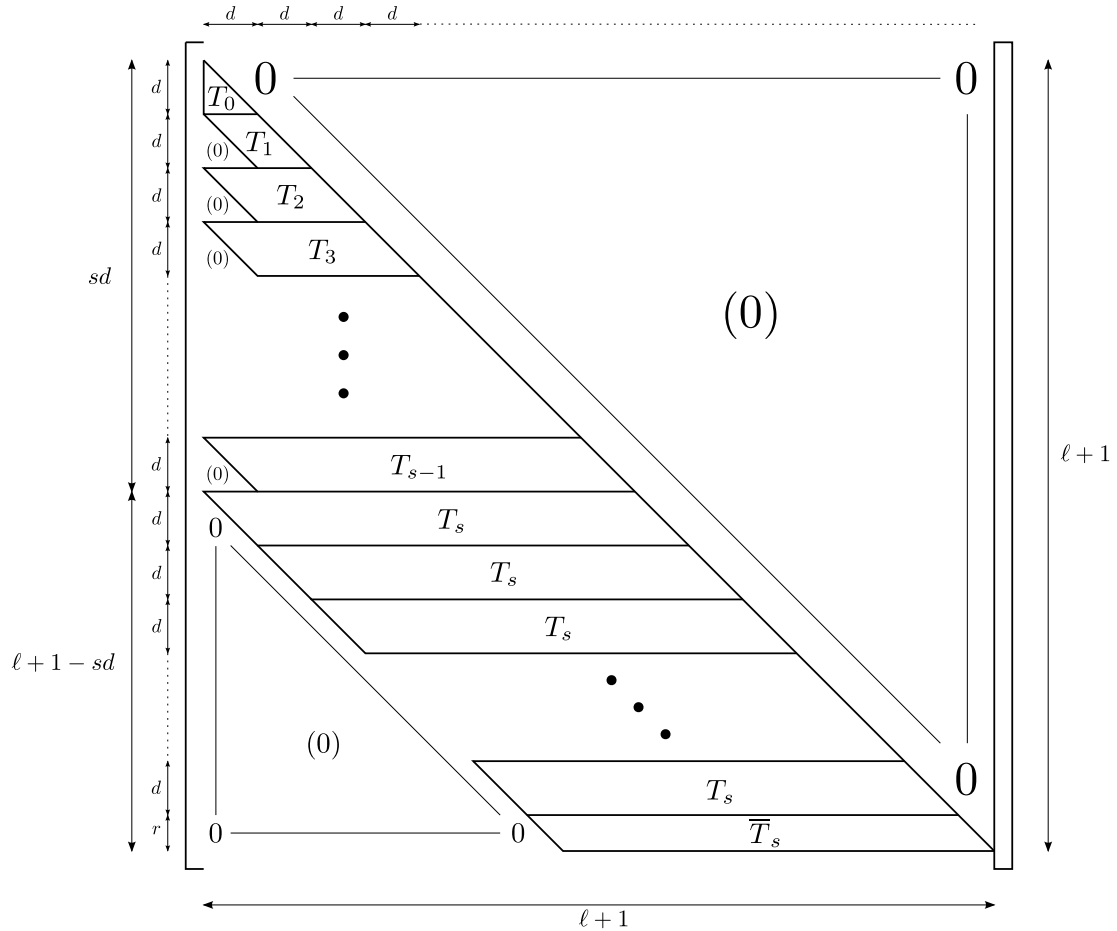
Find a solution of a structured linear system,



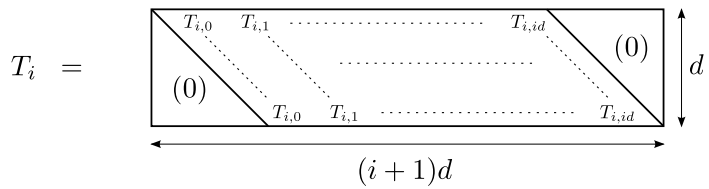
where $A_{i,j}$ is a Toeplitz / Hankel / Vandermonde / ... matrix

AriC — (list-)Decoding

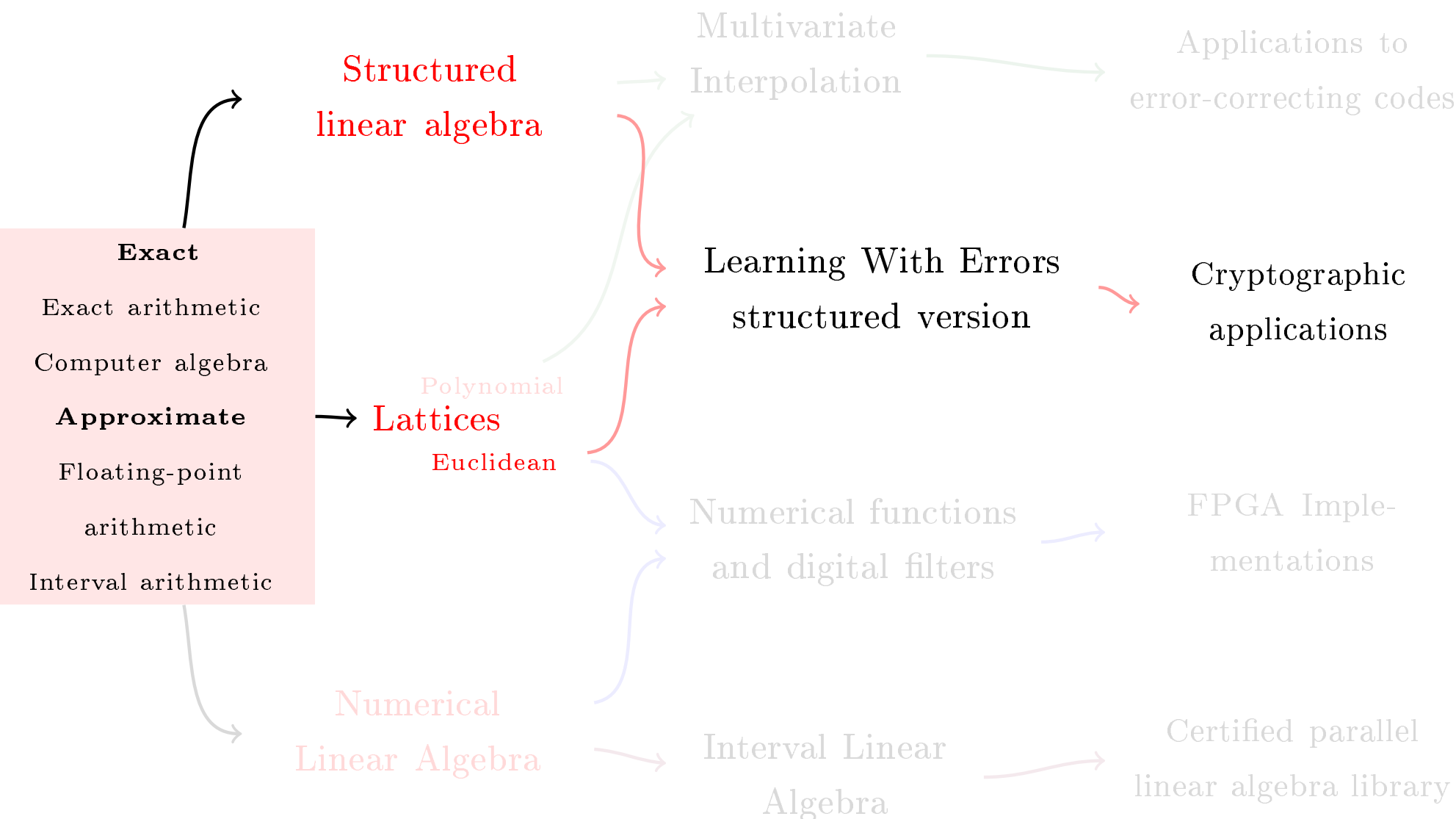
Find a short vector in a (structured) polynomial lattice,



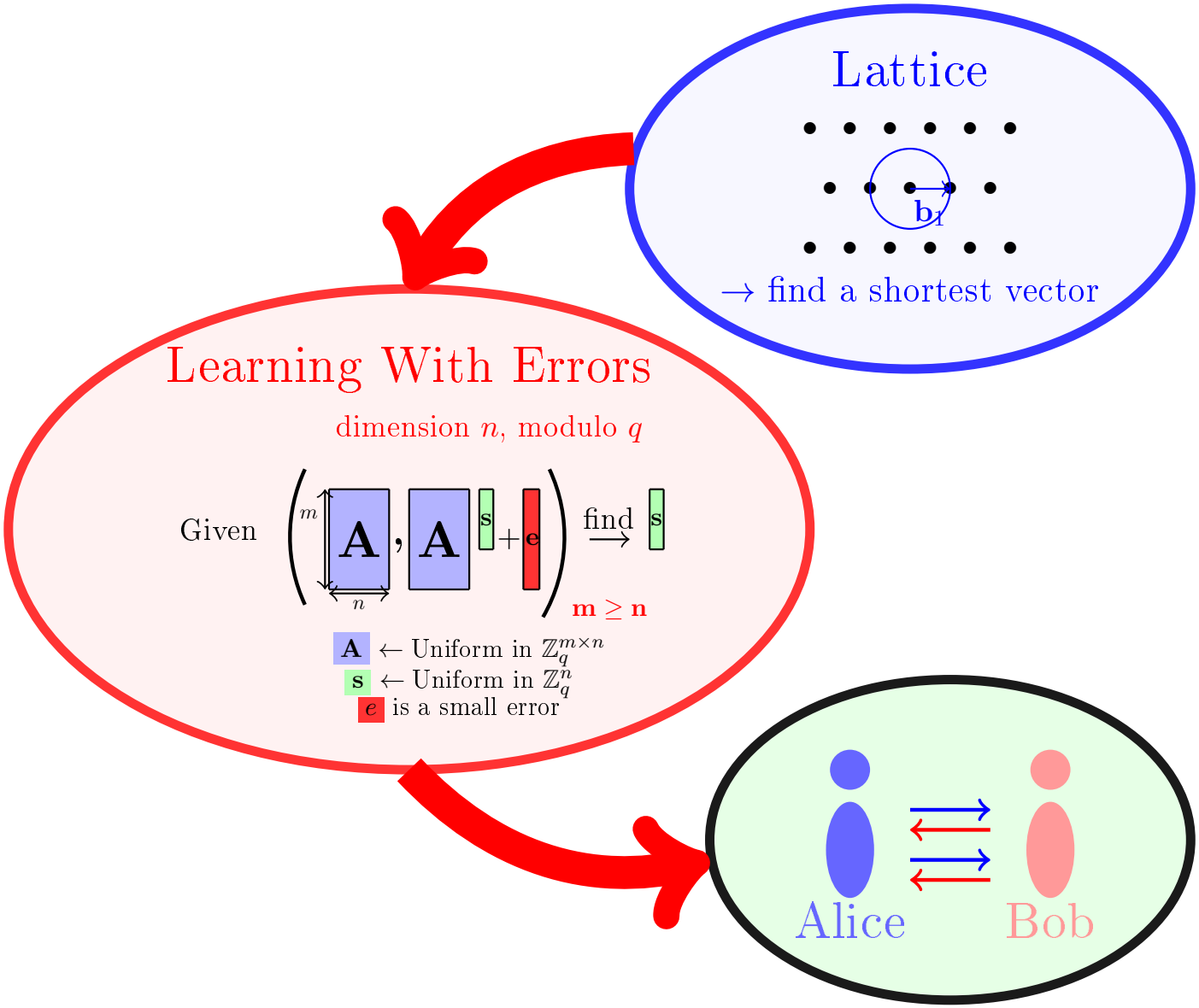
where T_i has a **Toeplitz** structure:



AriC


Structured
linear algebra
Multivariate
Interpolation
Applications to
error-correcting codes
Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point
arithmetic
Interval arithmetic
Polynomial
Lattices
Euclidean
Learning With Errors
structured version
Cryptographic
applications
Numerical functions
and digital filters
FPGA Imple-
mentations
Numerical
Linear Algebra
Interval Linear
Algebra
Certified parallel
linear algebra library

AriC – Lattice-Based Cryptography



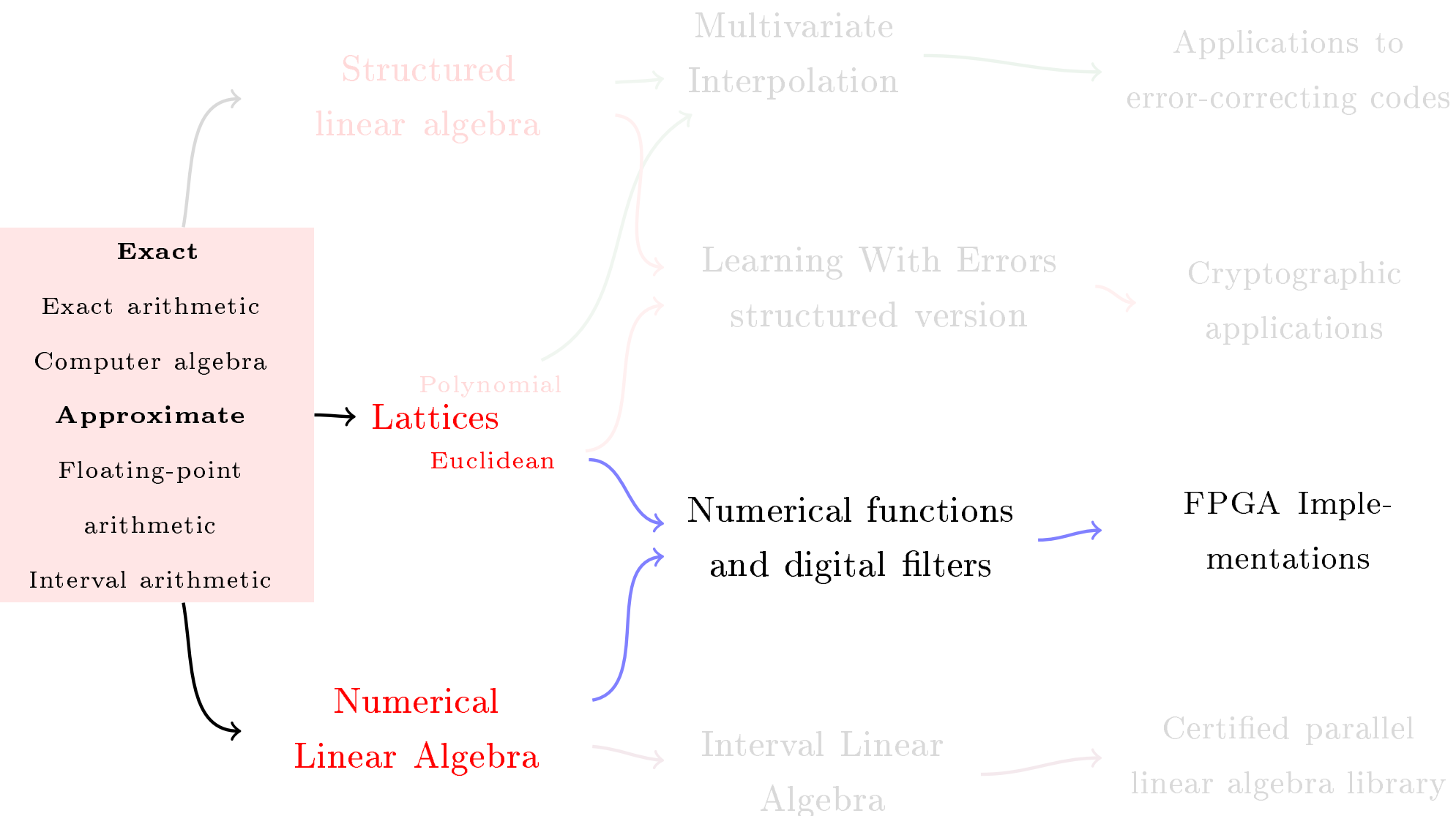
AriC – Lattice-Based Cryptography

- Public Key Encryption
- Identity Based Encryption
- Fully Homomorphic Encryption

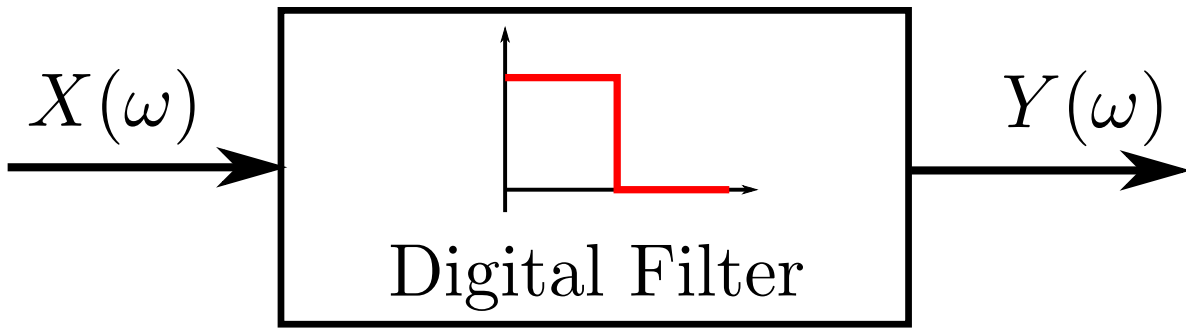
- Signature
- Group Signature
- Hash Function

- Cryptographic Multilinear Maps

AriC


Structured
linear algebra
Multivariate
Interpolation
Applications to
error-correcting codes
Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point
arithmetic
Interval arithmetic
Learning With Errors
structured version
Cryptographic
applications
Polynomial
Lattices
Euclidean
Numerical functions
and digital filters
FPGA Imple-
mentations
Numerical
Linear Algebra
Interval Linear
Algebra
Certified parallel
linear algebra library

AriC – Digital Filter Design



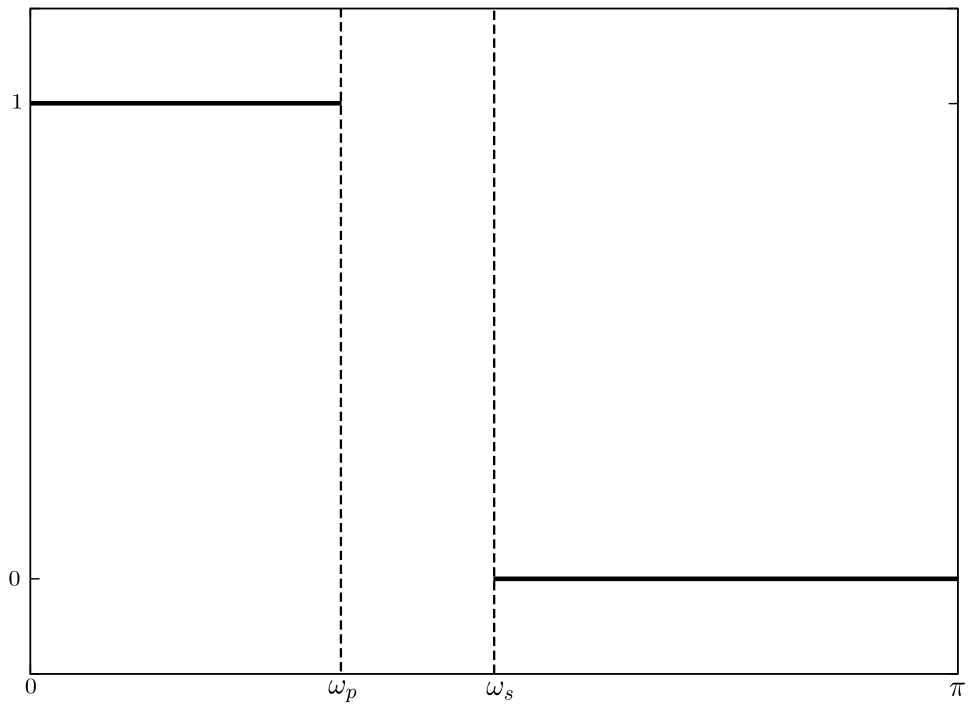
$$Y(\omega) = H_d(\omega)X(\omega), \omega \in [0, \pi]$$

Two types of filters:

- finite impulse response (**FIR**) $\Rightarrow$ $H_d(\omega)$ **polynomial**
- infinite impulse response (**IIR**) $\Rightarrow$ $H_d(\omega)$ **rational function**

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

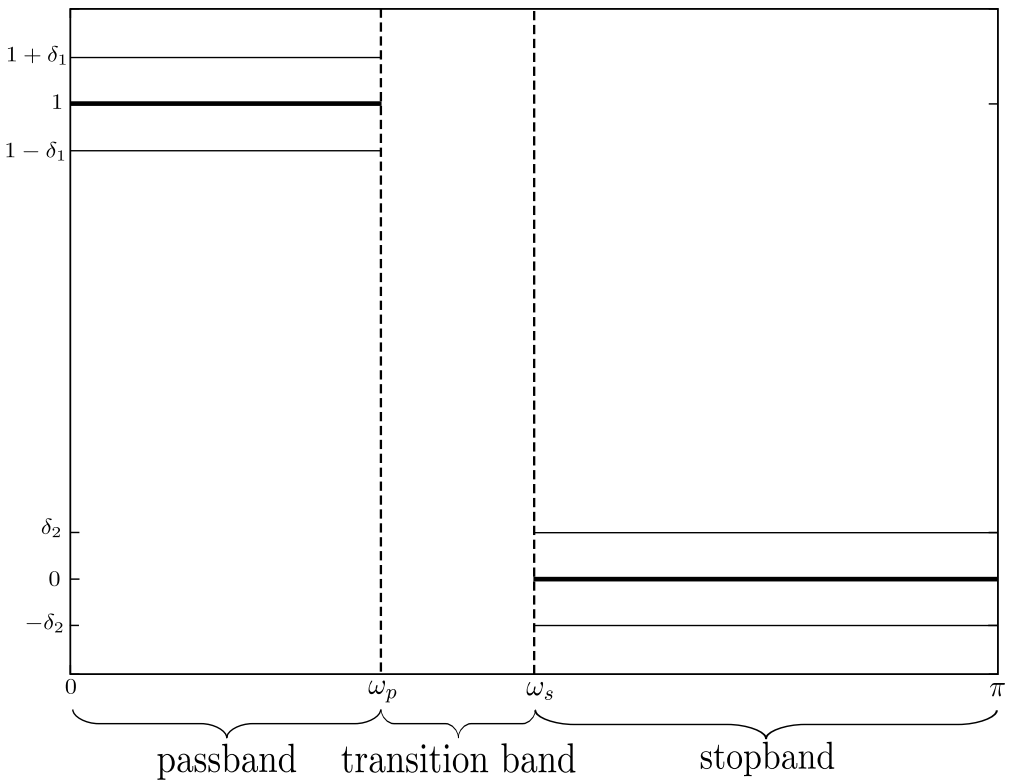
2. Coefficient quantization:

$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

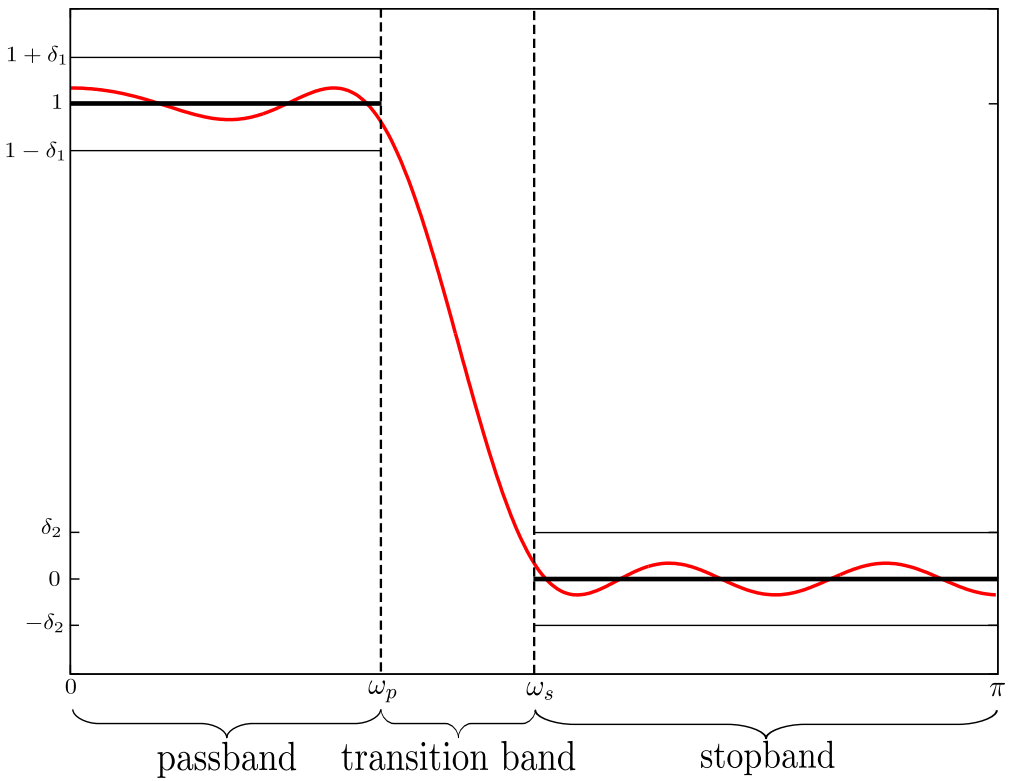
2. Coefficient quantization:

$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

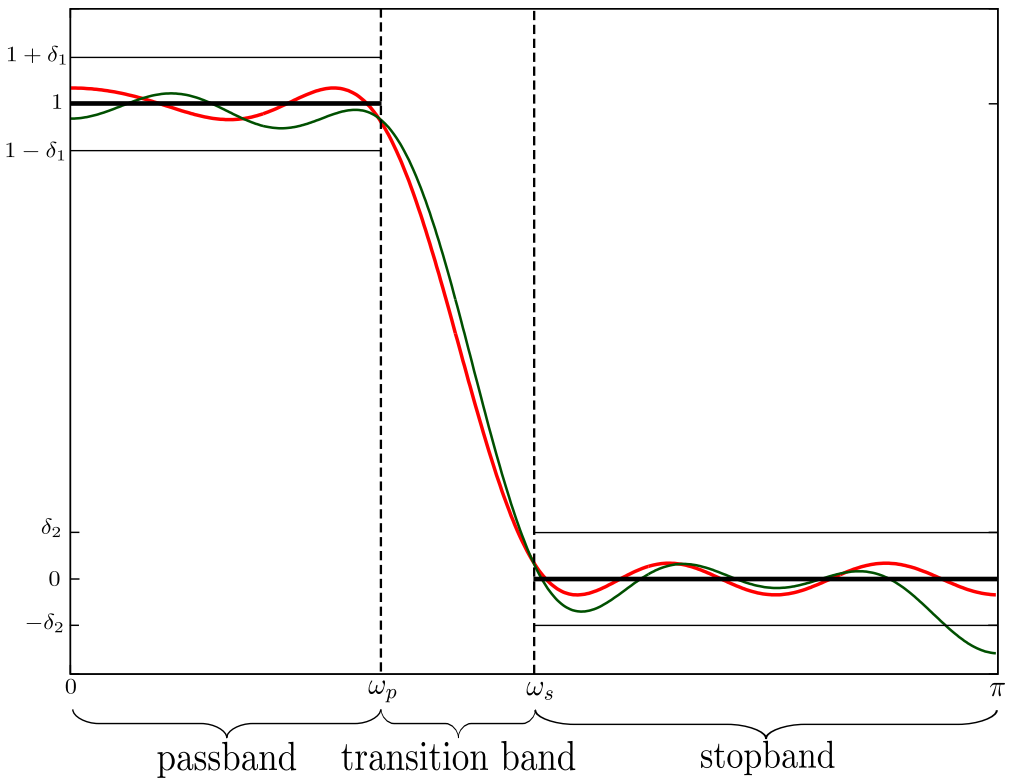
2. Coefficient quantization:

$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

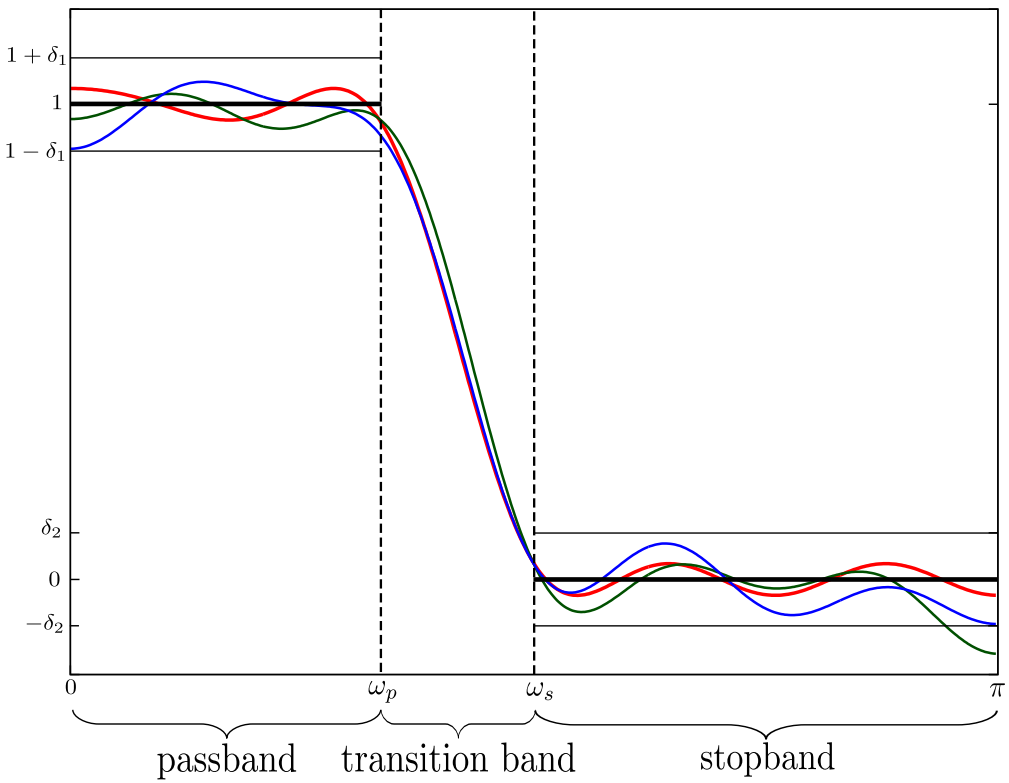
2. Coefficient quantization:

$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

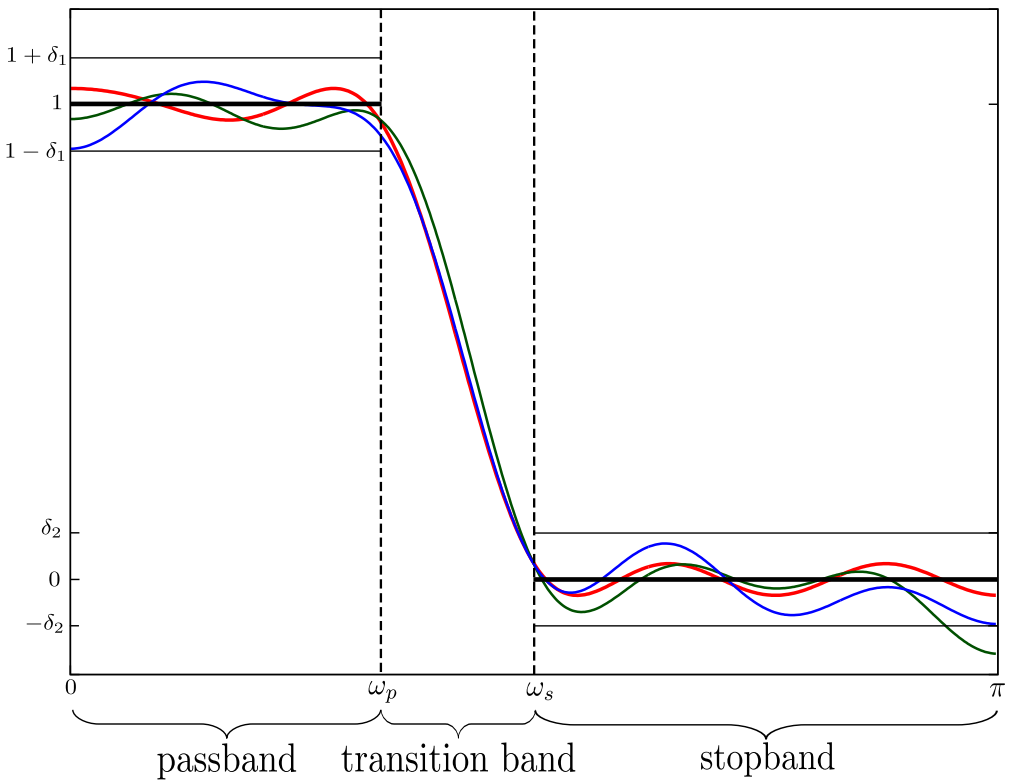
2. Coefficient quantization:

$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC – Digital Filter Design

FIR case: $H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$



Steps:

1. Optimal filter computation:

$$H_d(\omega) = \sum_{k=0}^{L} a_k \cos(\omega k)$$

Naive rounding:

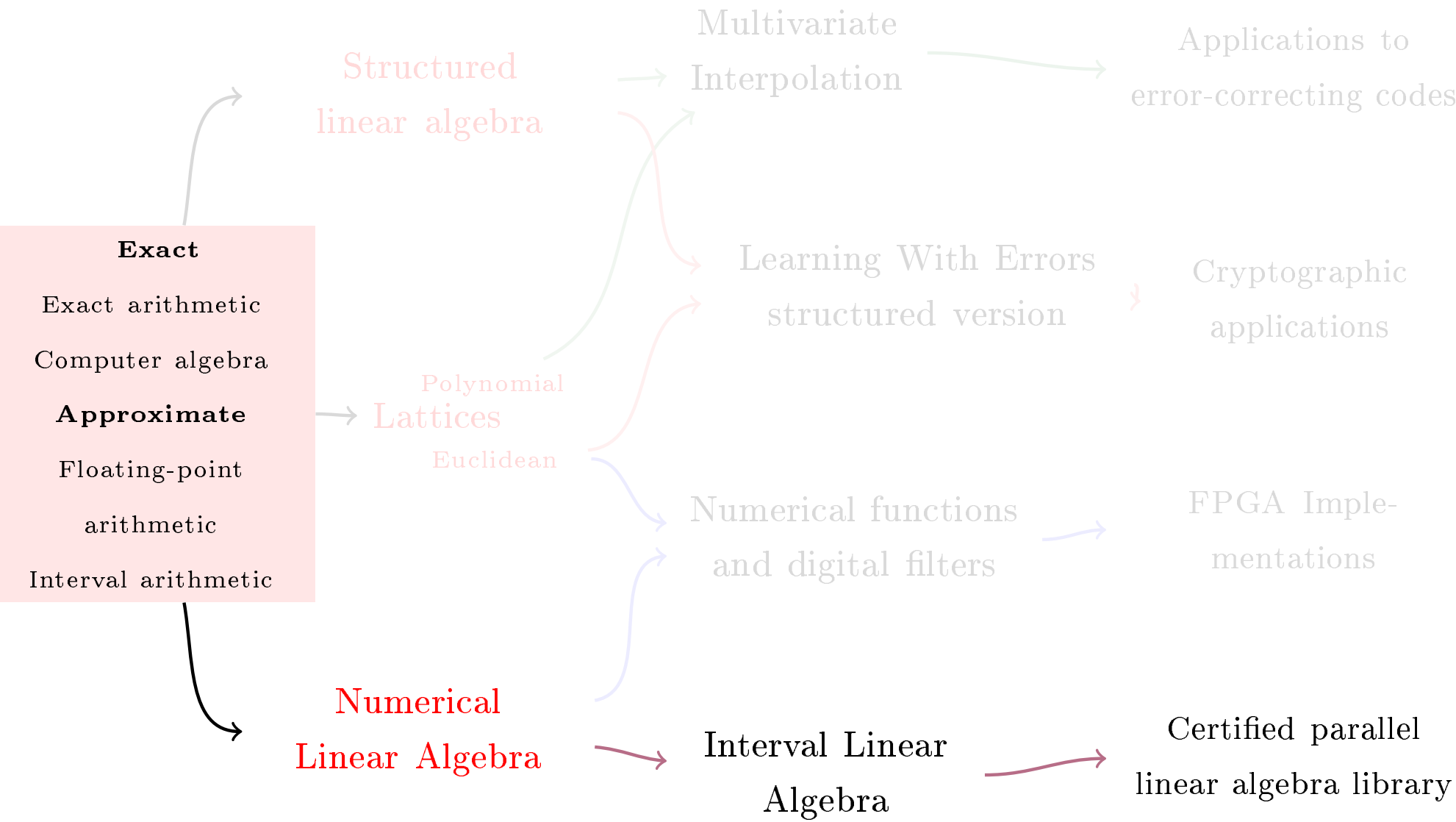
$$\overline{H}_d(\omega) = \sum_{k=0}^{L} \overline{a}_k \cos(\omega k)$$

2. Coefficient quantization:

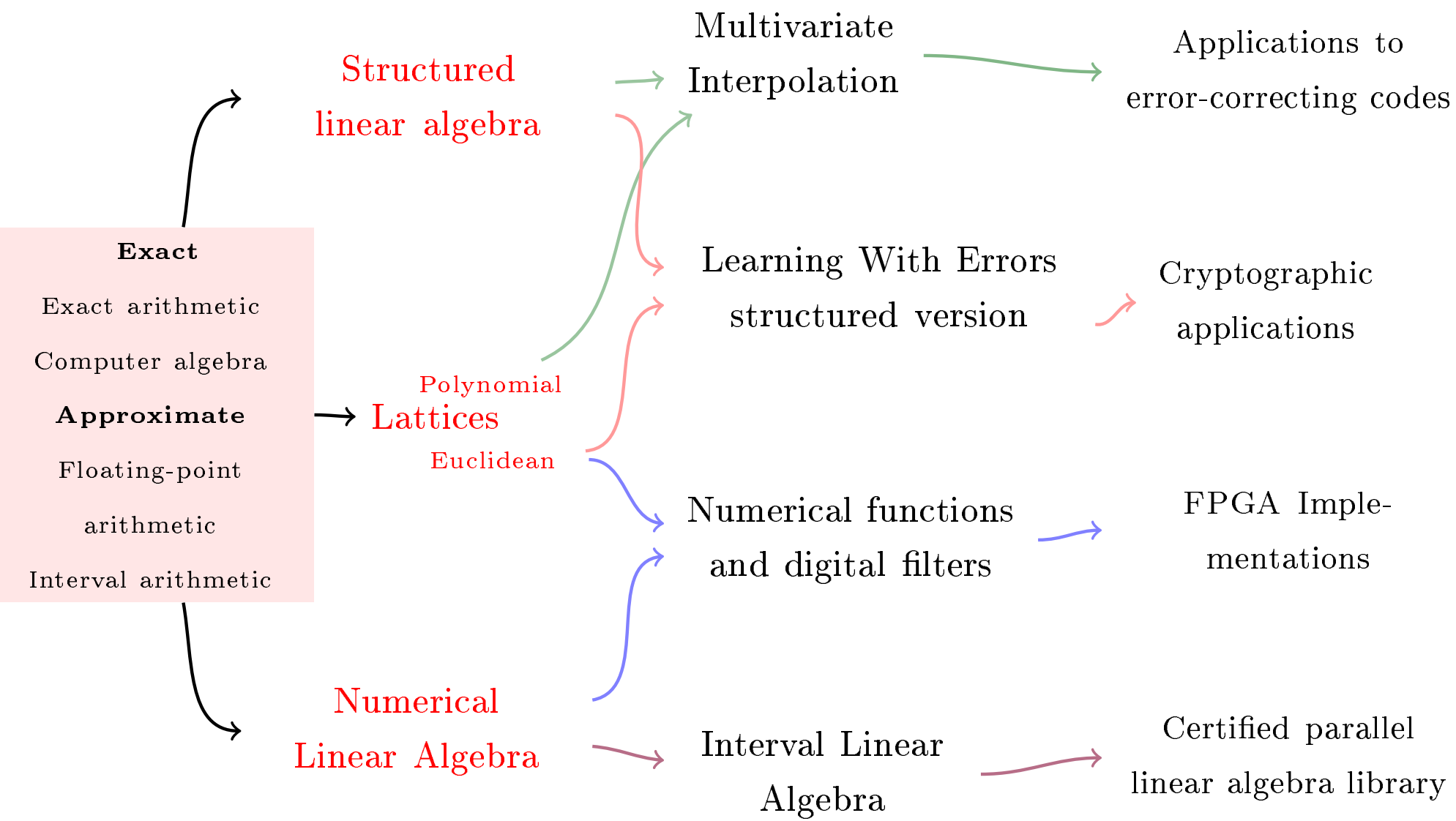
$$H_d^*(\omega) = \sum_{k=0}^{L} a_k^* \cos(\omega k)$$

Goal: filter synthesis toolchain for embedded and FPGA targets

AriC


Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point arithmetic
Interval arithmetic
Structured linear algebra
Polynomial
Lattices
Euclidean
Numerical Linear Algebra
Multivariate Interpolation
Learning With Errors structured version
Numerical functions and digital filters
Interval Linear Algebra
Applications to error-correcting codes
Cryptographic applications
FPGA Implementations
Certified parallel linear algebra library

AriC


Exact
Exact arithmetic
Computer algebra
Approximate
Floating-point
arithmetic
Interval arithmetic
Structured
linear algebra
Polynomial
Lattices
Euclidean
Numerical
Linear Algebra
Multivariate
Interpolation
Learning With Errors
structured version
Numerical functions
and digital filters
Interval Linear
Algebra
Applications to
error-correcting codes
Cryptographic
applications
FPGA Imple-
mentations
Certified parallel
linear algebra library