

# Introduction to Spring School on Polyhedral Code Analysis and Optimizations

Alain Darte

CNRS, Compsys  
Laboratoire de l'Informatique du Parallélisme  
École normale supérieure de Lyon

May 13-17, 2013, St Germain au Mont d'Or



# Outline

- 1 Presentation of the school
  - Participants
  - Labex thematic quarter
  - Organization principles
- 2 The polyhedral model
  - Analyses, optimizations, and tools
  - The polyhedral model is... a model
- 3 Topics, courses, and other aspects
  - History
  - Courses
  - Organization details

# Welcome to all participants!

55 participants.

## Origin

France (25), Germany (8), USA (8), UK (4), Netherlands (3), Spain (3), China (1), India (1), Ireland (1), Italy (1).

## Status

PhD students (33), Post-doc (3), Academic (16), Industry (3).

## Main groups

- Some well-identified “polyhedral” groups in France, Germany, US, Netherlands, UK, and industry (Reservoir Labs, Silkan).
- Interestingly, some – a priori – more distant groups w.r.t. the polyhedral model in Spain, UK, Ireland, Italy, China, industry (ARM).
- Some important contributors who could not come, e.g., Risset, Griehl, Loechner, Clauss, Ancourt, Bastoul, Derrien, Wonnacott, Ramanujam, Catthoor, Xue, Kienhuis, Grösslinger, etc.

☛ Don't forget your one-page pdf file with keywords & picture!

# Labex MILYON and labex compilation

Labex is a national funding mechanism to structure research at the regional level, through a national loan for investment.

## Labex MILYON and labex compilation

**Labex** is a national funding mechanism to structure research at the regional level, through a national loan for investment.

**Labex MILYON** to federate computer science and mathematics in Lyon, for research, teaching, industrial transfer, to increase international exchanges, mainly through **thematic quarters**.

# Labex MILYON and labex compilation

**Labex** is a national funding mechanism to structure research at the regional level, through a national loan for investment.

**Labex MILYON** to federate computer science and mathematics in Lyon, for research, teaching, industrial transfer, to increase international exchanges, mainly through **thematic quarters**.

## Thematic quarter on compilation

- April 2-4: French compilation days.
- May 13-17: School on polyhedral code analysis & opt.
- June 28-July 2: Keynotes for a dive into HPC languages.
- July 3-5: CPC'13, 17th workshop on compilers for parallel computing

Check out <http://labexcompilation.ens-lyon.fr>

Many thanks to Alexandre Isoard, Laetitia Lécot, Sophie Azzaro.

# Principles of the spring school

## Goals

- Complementary to Impact workshop (new results).
- Teach polyhedral world to new **PhD students** or **researchers**.
- Try to agree on a **state-of-the art** for our community.
- First school of a series? Book? Web site? Wikipedia?

# Principles of the spring school

## Goals

- Complementary to Impact workshop (new results).
- Teach polyhedral world to new **PhD students** or **researchers**.
- Try to agree on a **state-of-the art** for our community.
- First school of a series? Book? Web site? Wikipedia?

## Organization principles

- Closed placed (in Dagstuhl spirit), cheap, limited (35 rooms).
- **7 half-day courses** + possibly advanced topics.
- 1 half-day of social event (walk + winery + “banquet”).



# Principles of the spring school

## Goals

- Complementary to Impact workshop (new results).
- Teach polyhedral world to new **PhD students** or **researchers**.
- Try to agree on a **state-of-the art** for our community.
- First school of a series? Book? Web site? Wikipedia?

## Organization principles

- Closed placed (in Dagstuhl spirit), cheap, limited (35 rooms).
- **7 half-day courses** + possibly advanced topics.
- 1 half-day of social event (walk + winery + “banquet”).

## Program

- Introduction (Darte), history (Rajopadhye), basics (Feautrier)
- Transf. & scheduling (Pouchet)
- Modeling (Verdoolaege)
- Abstract interpretation (Miné)
- Region analysis (Creusillet)
- Distr. memory (Bondhugula)
- SIMD (Sadayappan/Vasilache)

# Outline

- 1 Presentation of the school
  - Participants
  - Labex thematic quarter
  - Organization principles
- 2 **The polyhedral model**
  - Analyses, optimizations, and tools
  - The polyhedral model is... a model
- 3 Topics, courses, and other aspects
  - History
  - Courses
  - Organization details

# Polyhedral model: the all-affine fully-analyzable world



## Fragments of Fortran DO loops:

```
...  
DO i=1,N  
  DO j=1,N  
    a(i,j) = c(i,j-1)  
    c(i,j) = a(i,j) + a(i-1,N)  
  ENDDO  
ENDDO  
...
```

- Nested loops, static control.
- **Iteration vector** and domain.
- Loop increment = 1.
- Affine bounds of surrounding counters and of **parameters**.
- Multi-dimensional arrays, same restriction for access functions.

# Polyhedral model: the all-affine fully-analyzable world



## Fragments of Fortran DO loops:

...

```
DO i=1,N
  DO j=1,N
    a(i,j) = c(i,j-1)
    c(i,j) = a(i,j) + a(i-1,N)
  ENDDO
ENDDO
...
```

- Nested loops, static control.
- **Iteration vector** and domain.
- Loop increment = 1.
- Affine bounds of surrounding counters and of **parameters**.
- Multi-dimensional arrays, same restriction for access functions.

## ☛ Exact element-wise analysis. Operations instead of statements.

- Iteration domain: polytope. Compact & regular representation.
- Sequential order  $\leq_{seq}$ : affine inequalities + unions.
- Data accesses: affine functions/relations on polytopes.
- **Parameters**, key feature for optimality, structure, complexity...

## (Parametric) analysis, transformations, optimizations

### Data-flow array analysis

- Array expansion.
- Single assignment.
- Liveness array analysis.
- Data reuse.

### Mapping computations & data

- Systolic arrays design.
- Data distribution.
- Communication opt.

### Loop transformations

- Automatic parallelization.
- Transformations framework.
- Code generation (with loops or with automaton).

### Counting & Ehrhart polynomials

- Cache misses.
- Memory size computations.
- Termination (e.g., WCET).

And many more...



## Typical criticism: such codes do not exist

No such codes? No interest? We believe it is **wrong**:

- Applicable to specific domains: e.g., signal/video processing.
- Required for **static automation**, very suitable for HLS.
- Can be limited to parts to analyze (SCoP, static control part).
- Central model & **source of inspiration** for more general cases.



# Typical criticism: such codes do not exist

No such codes? No interest? We believe it is **wrong**:

- Applicable to specific domains: e.g., signal/video processing.
- Required for **static automation**, very suitable for HLS.
- Can be limited to parts to analyze (SCoP, static control part).
- Central model & **source of inspiration** for more general cases.

Two main phases of developments of the polyhedral model

- 90s: mainly theory, methodologies, algorithms, limits.
- 2000-: new generation, make it work, develop robust tools.



# Typical criticism: such codes do not exist

No such codes? No interest? We believe it is **wrong**:

- Applicable to specific domains: e.g., signal/video processing.
- Required for **static automation**, very suitable for HLS.
- Can be limited to parts to analyze (SCoP, static control part).
- Central model & **source of inspiration** for more general cases.

Two main phases of developments of the polyhedral model

- 90s: mainly theory, methodologies, algorithms, limits.
- 2000-: new generation, make it work, develop robust tools.

## Tools

- **Pip**: parametric (I)LP.
- **Polylib & Isl**: polyhedra and sets.
- **Fadalib**: fuzzy data-flow analysis.
- **Cloog**: code generation.
- **Ehrhart & Barvinok**: counting.
- **Cl@k**: critical lattices.

## Compiler or infrastructures

- **Alpha/Hard**: SAREs to HLS.
- **Graphite**: library for GCC.
- **Gecos & Chuba**: tools for HLS.
- **Pips4all, Pluto & R-Stream**: parallelizing compilers.
- **Compaan**: polyhedral streams.



## Different language variants fit the polyhedral model

### C for loops:

```
for (i=1, i<=N, i++) {  
  for (j=1, j<=N, j++) {  
    a[i][j] = c[i][j-1];  
    c[i][j] = a[i][j] + a[i-1][N];  
  }  
}
```

### Uniform recurrence equations

$\forall(i, j)$  such that  $1 \leq i, j \leq N$

$$\begin{cases} a(i, j) = c(i, j - 1) \\ b(i, j) = a(i - 1, j) + b(i, j + 1) \\ c(i, j) = a(i, j) + b(i, j) \end{cases}$$

### C while loops:

```
y = 0; x = 0;  
while (x <= N && y <= N) {  
  if (?) {  
    x=x+1;  
    while (y >= 0 && ?) y=y-1;  
  }  
  y=y+1;  
}
```

### FAUST: audio processing

```
random = +(12345) ~ *(1103515);  
noise = random/2147483.0;  
process = random/2 : @(10);
```

## Different language variants fit the polyhedral model

### C for loops:

```
for (i=1, i<=N, i++) {  
  for (j=1, j<=N, j++) {  
    a[i][j] = c[i][j-1];  
    c[i][j] = a[i][j] + a[i-1][N];  
  }  
}
```

### Uniform recurrence equations

$\forall(i,j)$  such that  $1 \leq i, j \leq N$

$$\begin{cases} a(i,j) = c(i,j-1) \\ b(i,j) = a(i-1,j) + b(i,j+1) \\ c(i,j) = a(i,j) + b(i,j) \end{cases}$$

### C while loops:

```
y = 0; x = 0;  
while (x <= N && y <= N) {  
  if (?) {  
    x=x+1;  
    while (y >= 0 && ?) y=y-1;  
  }  
  y=y+1;  
}
```

### FAUST: audio processing

```
random = +(12345) ~ *(1103515);  
noise = random/2147483.0;  
process = random/2 : @(10);
```

and more can benefit from the polyhedral model: Fortran90, HPF, Matlab, C for HLS, X10, OpenStream, CnC, ...

# But still, how to deal with non-static control programs?

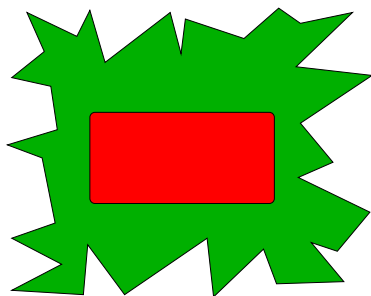
Polyhedral model.



# But still, how to deal with non-static control programs?

Polyhedral model.

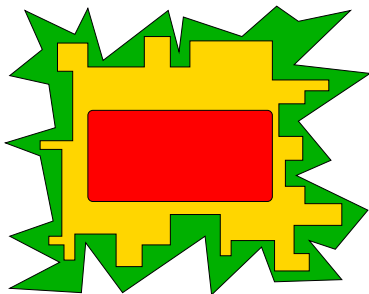
Real life.



# But still, how to deal with non-static control programs?

Polyhedral model.

Real life.



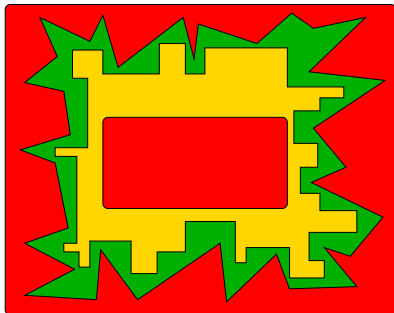
Extensions.

- Non-affine constraints.
- Handling of while loops.
- Recursive programs.
- Beyond induction variables.

# But still, how to deal with non-static control programs?

Polyhedral model.

Real life.



Extensions.

- Non-affine constraints.
- Handling of while loops.
- Recursive programs.
- Beyond induction variables.

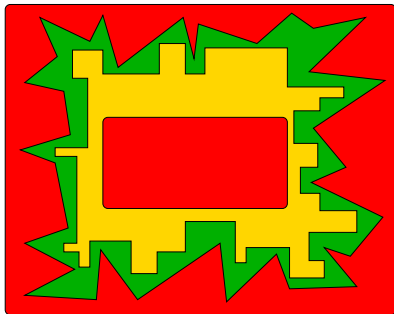
Approximations.

- Dependences, lifetime, data & iteration domains, etc.
- **Do not assume exact information is available.**

# But still, how to deal with non-static control programs?

Polyhedral model.

Real life.



Extensions.

- Non-affine constraints.
- Handling of while loops.
- Recursive programs.
- Beyond induction variables.

Approximations.

- Dependences, lifetime, data & iteration domains, etc.
- **Do not assume exact information is available.**

Think conservative!

☛ Link with abstract interpretation & array region analysis

# Outline

- 1 Presentation of the school
  - Participants
  - Labex thematic quarter
  - Organization principles
- 2 The polyhedral model
  - Analyses, optimizations, and tools
  - The polyhedral model is... a model
- 3 Topics, courses, and other aspects
  - History
  - Courses
  - Organization details

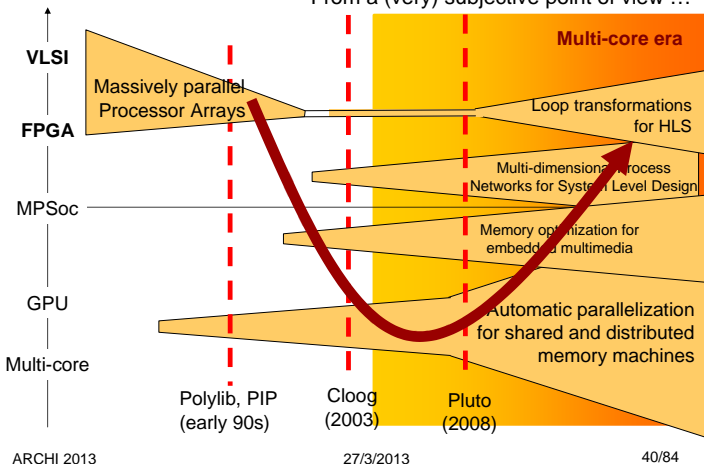


# One view of history (slide borrowed from Steven Derrien)

## A short story of the polyhedral model



From a (very) subjective point of view ...



## Main courses

Most courses by contributors from the “new generation”.

[Sanjay Rajopadhye](#) A view on history.

[Paul Feautrier](#) “Basics” in terms of mathematical concepts.

[Louis-Noël Pouchet](#) Loop transformations, scheduling.

[Sven Verdoolaege](#) Modeling analysis and optimizations.

[Antoine Miné](#) Abstract interpretation.

[Béatrice Creusillet](#) Array region analysis.

[Uday Bondhugula \(+ myself\)](#) Compilation for distributed memory.

[Sadayappan \(+ Nicolas Vasilache\)](#) SIMD architectures.

Note: hard work although, by nature, biased, incomplete, redundant!

# Main courses

Most courses by contributors from the “new generation”.

[Sanjay Rajopadhye](#) A view on history.

[Paul Feautrier](#) “Basics” in terms of mathematical concepts.

[Louis-Noël Pouchet](#) Loop transformations, scheduling.

[Sven Verdoolaege](#) Modeling analysis and optimizations.

[Antoine Miné](#) Abstract interpretation.

[Béatrice Creusillet](#) Array region analysis.

[Uday Bondhugula \(+ myself\)](#) Compilation for distributed memory.

[Sadayappan \(+ Nicolas Vasilache\)](#) SIMD architectures.

Note: hard work although, by nature, biased, incomplete, redundant!

Order should remain fix but schedule may change w.r.t. weather.

Monday: ☁ Tuesday: ☀ Wednesday: ☁ Thursday: ☁ Friday: ☁

## Additional talks and discussions

Many topics are missing, e.g., tiling, dependence analysis, induction variable recognition, array privatization, loop fusion, optimality, code generation, Ehrhart theory, locality optimizations, benchmarks, high-level synthesis, trace analysis, program equivalence, termination, extensions, pipelining, streams, . . . **Next time?**

## Additional talks and discussions

Many topics are missing, e.g., tiling, dependence analysis, induction variable recognition, array privatization, loop fusion, optimality, code generation, Ehrhart theory, locality optimizations, benchmarks, high-level synthesis, trace analysis, program equivalence, termination, extensions, pipelining, streams, ... **Next time?**

Additional talks are possible (but when?):

Patrice Quinton (Tue. pm?) Modularity in the polyhedral model.

Myself? KMW: scheduling versus computability

Vladimir Klebanov? Software verification?

...

## Additional talks and discussions

Many topics are missing, e.g., tiling, dependence analysis, induction variable recognition, array privatization, loop fusion, optimality, code generation, Ehrhart theory, locality optimizations, benchmarks, high-level synthesis, trace analysis, program equivalence, termination, extensions, pipelining, streams, ... **Next time?**

Additional talks are possible (but when?):

Patrice Quinton (Tue. pm?) Modularity in the polyhedral model.

Myself? KMW: scheduling versus computability

Vladimir Klebanov? Software verification?

...

How to collect comments, criticisms, references? Slides? Videos?

## Additional organization details

- Exact schedule.
- Rooms upstairs.
- Specific meals.
- Departures.
- Rooms on Friday.
- Tuesday (?): village + walk + village + winery.
- Wednesday: aperitif + “banquet” + magic.
- Walks, bikes, rides, . . . but rain.
- Any requests (scientific or organization).

## Additional organization details

- Exact schedule.
- Rooms upstairs.
- Specific meals.
- Departures.
- Rooms on Friday.
- Tuesday (?): village + walk + village + winery.
- Wednesday: aperitif + “banquet” + magic.
- Walks, bikes, rides, . . . but rain.
- Any requests (scientific or organization).

And now. . . let's work!