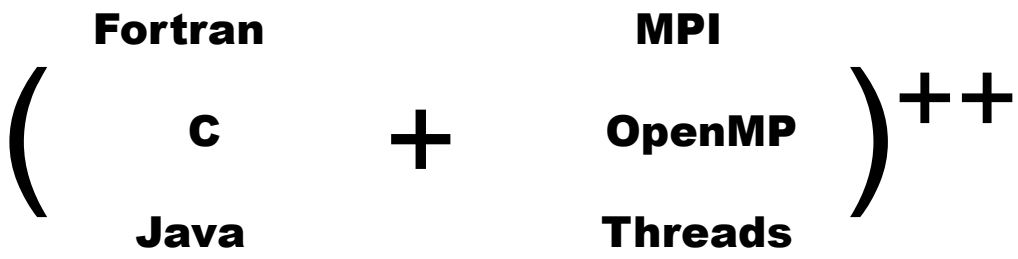
Advanced Stencil-Code Engineering (ExaStencils)

Christian Lengauer

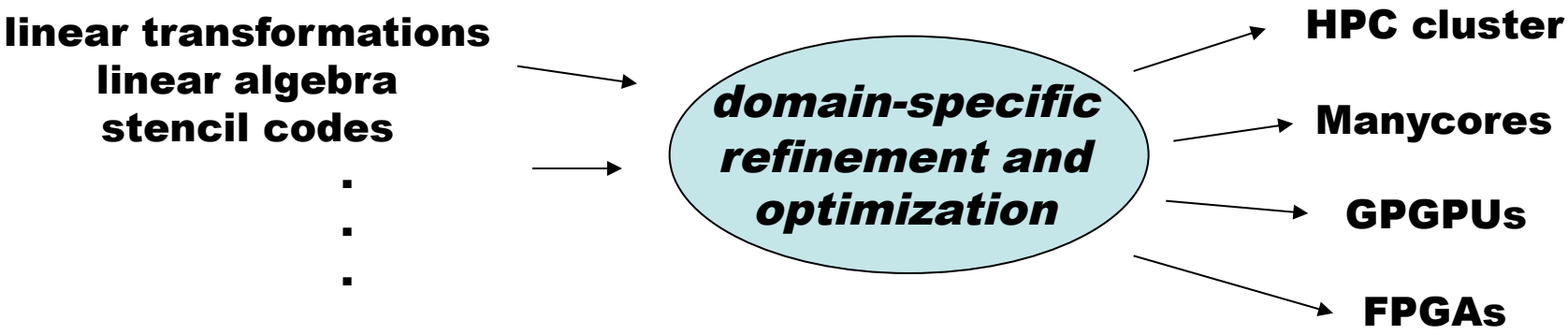
3rd Workshop on Extreme-Scale Programming Tools
New Orleans, 17.11.2014

Two Alternative Approaches in SPPEXA

- ***The evolutionary approach***



- ***The revolutionary approach***



Why is this Revolutionary?

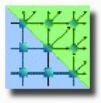
- *No general-purpose programming language*
  - Different refinement levels have their own domain-specific language
- *Exploitation of domain knowledge at all levels fo refinement*
  - About input data
  - About the algorithm
  - About the execution platform
- *Exploitation of common properties of programs*
  - Programs are not individuals but members of a family, a ”product line”
  - A product line specifies variabilities (so-called features)
  - Common properties and individual variations are stated explicitly and precisely
  - The ”programming” of a product is done by selection options (and nothing else!)
- *Still – the promise: full automation*
  - The target code is being ”weaved” automatically, optimized for the features selected
  - The optimization exploits knowledge about the specific feature combination

Our Project in the Radical Tier of SPPEXA

ExaStencils

http://www.exastencils.org/

A new, tool-assisted,
domain-specific
codesign approach for stencil codes



Jürgen Teich, Frank Hannig, Christian Schmitt

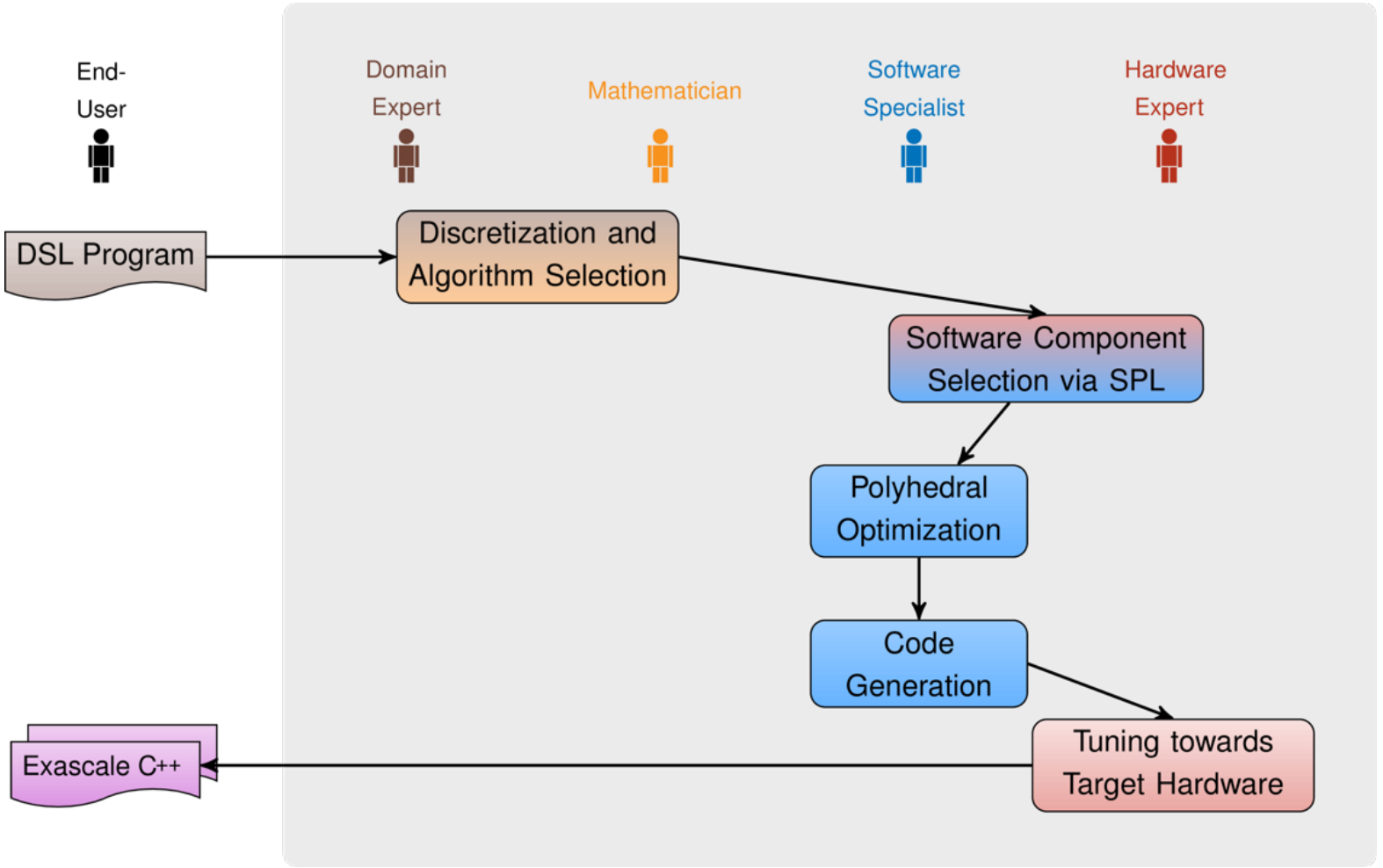
Ulrich Rüde, Harald Köstler, Sebastian Kuckuk

Matthias Bolten, Hannah Rittich

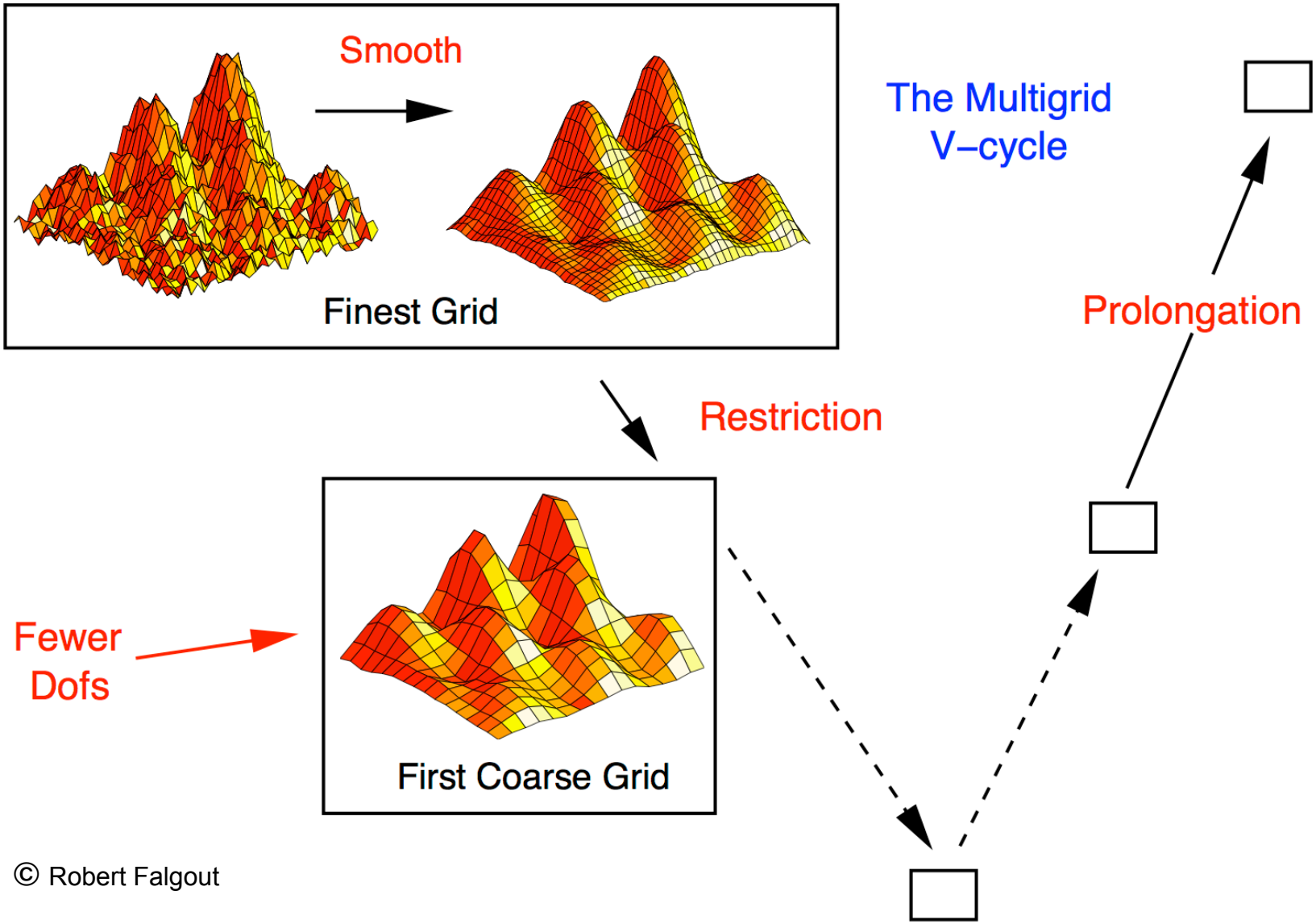
Christian Lengauer, Armin Größlinger, Stefan Kronawitter

Sven Apel, Alexander Grebhahn

Work Flow of ExaStencils

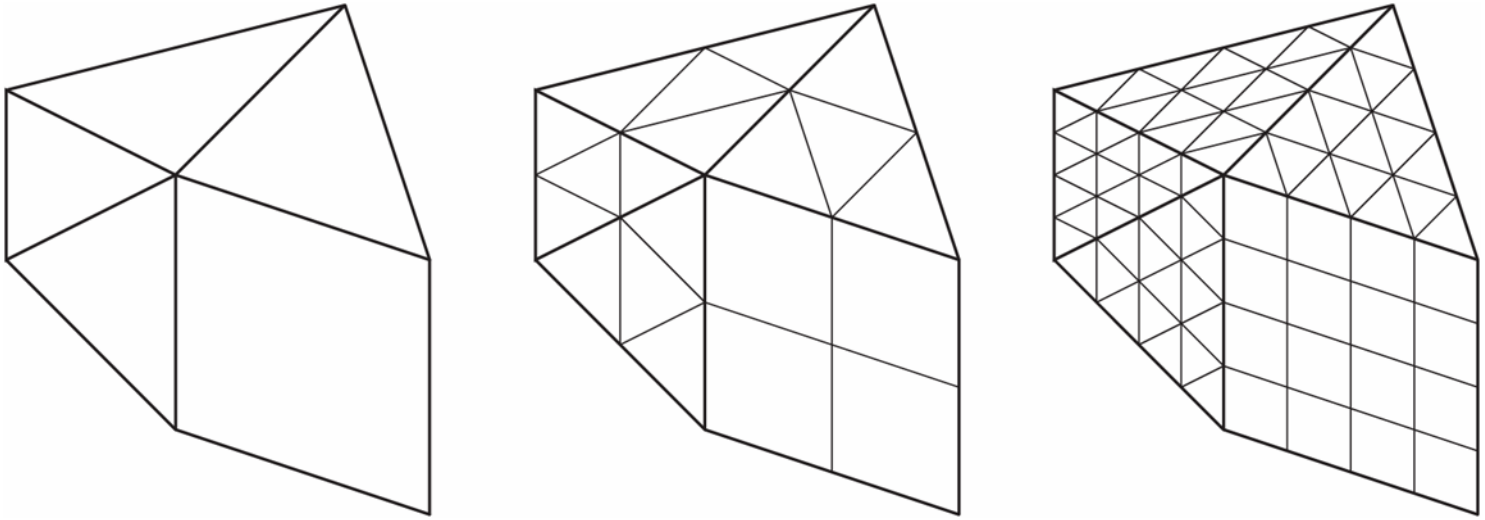


Domain: Multigrid Stencil Codes



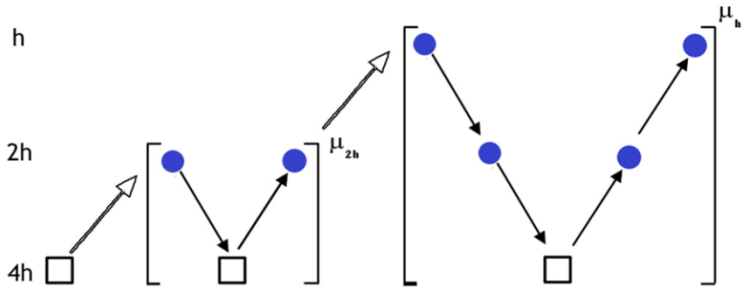
© Robert Falgout

Hierarchical Hybrid Grids



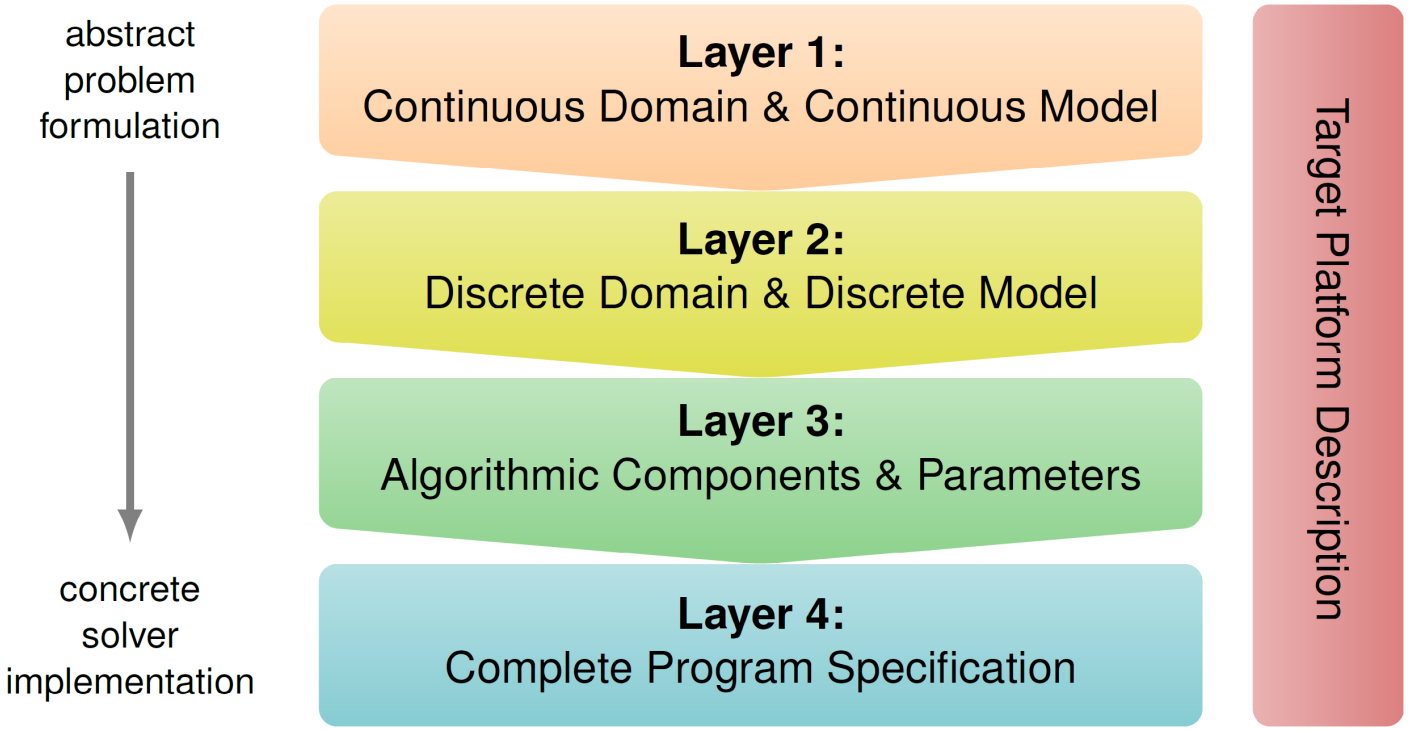
Algorithmic Engineering

- *Variabilities*
  - Discretization method
  - Grid transfer method
  - Cycling strategy
  - Smoother



- *Variants must reduce*
  - Convergence rate (platform-independent)
  - Execution time (platform-dependent)
- *Current activities*
  - A-priori prediction of the convergence rate by local Fourier analysis (LFA)
  - Extend LFA techniques to block-smoothers and aggressive coarsening

Domain-Specific Representation



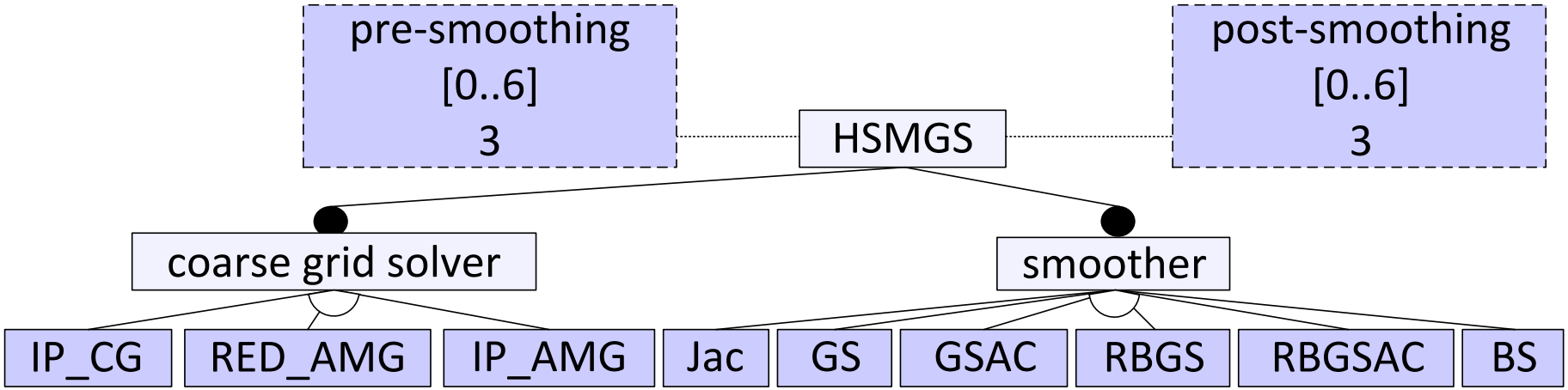
- *Current status*
  - Educated choice of Scala as the host language
  - Preliminary code generator for proof of concept finalized
  - Serious code generator for 80% of Layer 4 shows exascale potential

Domain-Specific Modelling and Optimization

- *Challenges*
  - Draft the software product line: identify the variabilities
  - Configuration options can interact in subtle ways (feature interaction)
  - Which combination of options gives the best performance?

- *Current status*
  - Adoption of techniques of automated software configuration
  - Design of a variability model for the Highly Scalable Multigrid Solver (HSMGS)
  - First experiments with a machine learning approach to identify efficient configurations
  - New: not only binary but also numerically parameterized options
  - Measurements of 10.2% of all variants → prediction accuracy of 89% on ave.
  - Still, so far, no domain-specific knowledge exploited!

Highly Scalable Multigrid Solver (HSMGS)



sum (pre-smoothing, post-smoothing) > 0

Legend:

IP_CG = In-Place Conjugate Gradient
IP_AMG = In-Place Algebraic multigrid
RED_AMG = Algebraic multigrid with data reduction
GSAC = Gauss-Seidel with additional communication
RBGSAC = Red-Black Gauss-Seidel with additional communication

Jac = Jacobi
GS = Gauss-Seidel
RBGS = Red-Black Gauss-Seidel
BS = Block-Smoother

Preliminary Code Generator

Variability	Layer	Options
Computational domain	DSL 1	UnitSquare, UnitCube
Operator	DSL 1	Laplacian, ComplexDiffusion
Boundary conditions	DSL 1	Dirichlet, Neumann
Location of grid points	DSL 2	node-based, cell-centered
Discretization	DSL 2	finite differences, finite volumes
Data type	DSL 2	single/double accuracy, complex numbers
Multigrid smoother	DSL 3	ω-Jacobi, ω-Gauss-Seidel, red-black variants
Multigrid inter-grid transfer	DSL 3	constant and linear interpolation and restriction
Multigrid coarsening	DSL 3	direct (re-discretization)
Multigrid parameters	DSL 3	various
Platform	Hardware	CPU, GPU
Parallelization	Hardware	serial, OpenMP

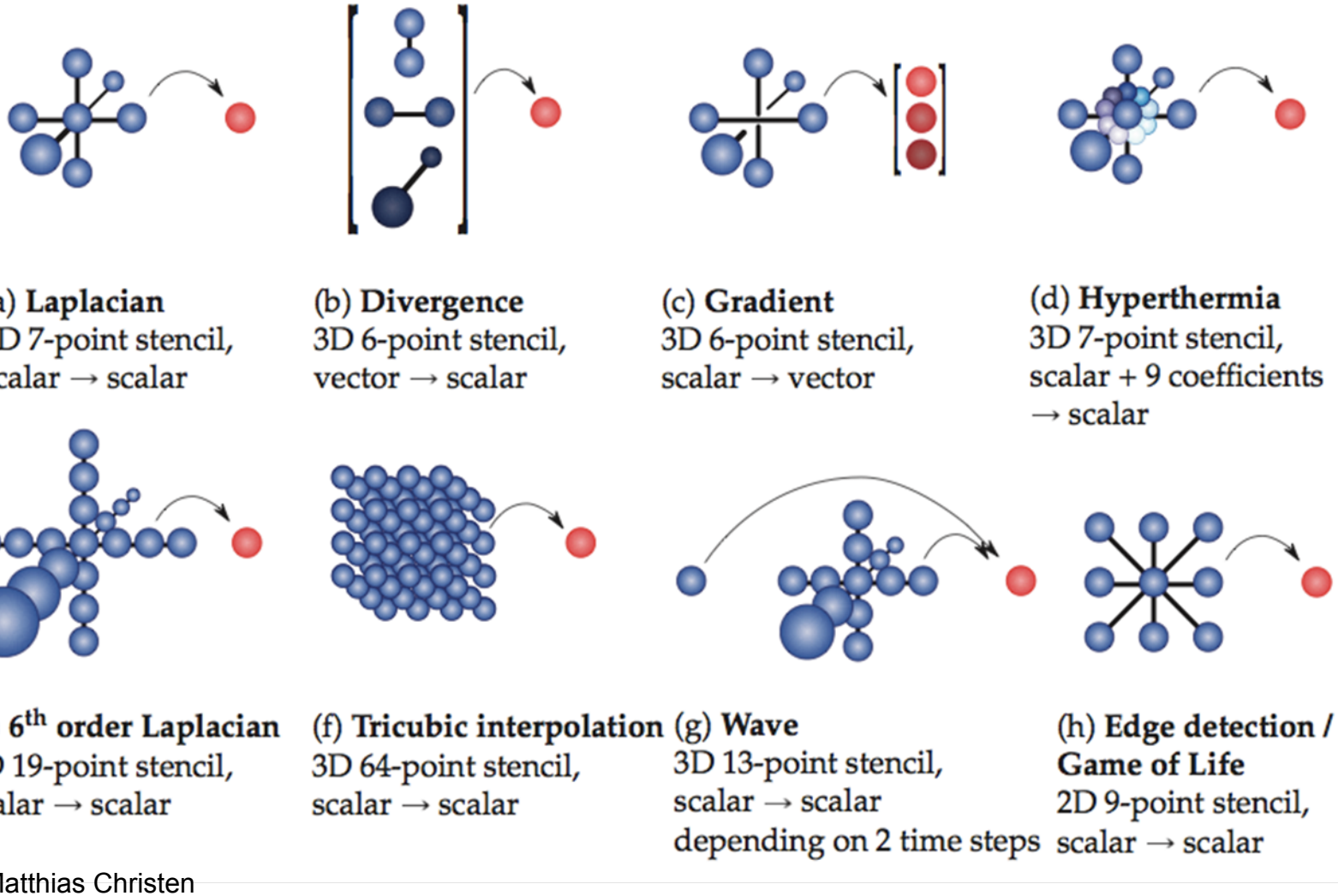
Novelties:

- Variant-driven code generation
- Wide spectrum of stencil codes

What Makes a Domain Suitable for the Radical Approach?

- *Size*
  - Considerably smaller than by contemporary expectations
- *Theoretical basis:*
  - Algebra
  - Conditional equations
- *Significance*
  - Stable abstract view
  - Stable, sustained user community
- *Examples*
  - FFTW: the fastest Fourier transform in the West
  - Spiral: discrete linear transforms
  - DBMSs: relational query optimization
  - cpp: Linux operating system configuration
  - ExaStencils: Multigrid stencil codes

Thanks for your Interest in the World of Stencils



(a) **Laplacian**
3D 7-point stencil,
scalar → scalar

(b) **Divergence**
3D 6-point stencil,
vector → scalar

(c) **Gradient**
3D 6-point stencil,
scalar → vector

(d) **Hyperthermia**
3D 7-point stencil,
scalar + 9 coefficients
→ scalar

(e) **6th order Laplacian**
3D 19-point stencil,
scalar → scalar

(f) **Tricubic interpolation**
3D 64-point stencil,
scalar → scalar

(g) **Wave**
3D 13-point stencil,
scalar → scalar
depending on 2 time steps

(h) **Edge detection / Game of Life**
2D 9-point stencil,
scalar → scalar