ESCAPE mini-applications in the service for common EULAG codebase

Zbigniew P. Piotrowski, Marcin Polkowski, Adam Ryczkowski, Farnoush Ataei

Institute of Meteorology and Water Management - National Research Institute

May 29th, 2018

This event is sponsored by the **"Numerical Weather Prediction for Sustainable Europe"** project, financed by the European fund within the **Smart Growth Operational Programme 2014–2020**, Measure 4.4. "Boosting human potential in R&D sector", through the **First Team 1/2016** award of **Foundation for Polish Science.**

European Funds Smart Growth

Foundation for Polish Science

European Union European Regional Development Fund

General motivation

- Due to the transfers of some experienced EULAG players to the European clubs (e.g. ECMWF Reading, IMGW Warsaw), EULAG codebase grows significantly slower than 5-15 years ago.
- Government funding, to large extent, shifts from the fundamental to applied research.
- Within the current resources in Academia, it is usually difficult to go beyond limited adapations of the EULAG legacy codebase ...
- ... yet the supercomputing and data-processing technology advances and it important to maintain state-of-the-art modeling and analysis capabilities.

Issues in the EULAG community

- The old code used to work for a particular case, but the newer does not, what has happened ? What am I doing wrong ?
- I want to try out the old code, but it does not compile on the modern hardware.
- I am getting pattern of processors in my model fields. Where is the bug ?
- What processor distribution/how many cores should I use for my computations ?
- I want to use these Eulerian and Lagrangian option together, but they share some scratch variables.
- I want to have history of a given diagnostic value at one point/column/region for the whole simulation.
- Could you please extract this part of the model for me so I can analyze that ?
- How do I integrate several equation sets at the same time ?

Proposed solution

To reengineer the EULAG codebase, to be able to

- share or use very similar codes for the basic research and applications,
- easily control the accuracy of physical implementation, e.g. does my code peform well in the already published benchmarks
- provide modern Fortran formulation, dispense with obsolete Fortran features, like common blocks
- abstract the memory allocation for the model variables from the scratch dataspace that can be reused in consecutive parts of the model
- automate verification of correctness of parallel implementation
- provide hints for the optimal processor distribution when given the number of cores

European research projects behind discusses advancements.

ESCAPE stands for Energy-efficient SCalable Algorithms for weather Prediction at Exascale project, funded by H2020-FETHPC-2014 call, currently in its final stage. The goal of the project is to "develop world-class, extreme-scale computing capabilities for European operational numerical weather prediction (NWP) and future climate models." It consists of 4 technical work packages:

- Weather and Climate Dwarfs (key components)
- Code Adaptation
- Hybrid Computing
- Benchmarking and Diagnostics

ESCAPE ideas are developed in the form of full weather dynamics solver within the "Numerical weather prediction for sustainable Europe" project carried out within the First Team programme of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund.

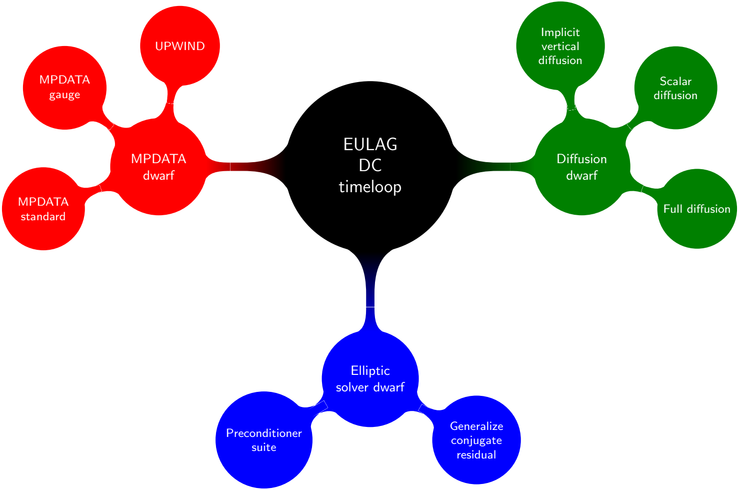
Selected design and implementation challenges for dynamical cores for weather and all-scale research

- Accurate implementation of numerical algorithms with proper boundary conditions.
- Building multi-level confidence, from unit-tests through numerical benchmarks, idealized testcases to objective NWP verification scores/comparison to measurements.
- Leveraging differential geometry apparatus for exposition of mathematical foundation of the numerical integration strategy.
- Taking advantage of "thin-atmosphere" 2D horizontal + 1D vertical formulation (similar splitting is observed within the modern supercomputing hardware !) while maintaining fully 3D turbulence characteristic of small scales.
- Exploiting energy-efficient supercomputing architectures up to the appropriate hardware limit

Interdisciplinary research motivated by ESCAPE

- NWP is constantly one of the most difficult challenges, and interdisciplinary specialists combining physics background, understanding of mathematics and skilled in computer science are rare.
- The usual answer in HPC is "use libraries", but these can only address selected technologies/algorithms and not necessarily can be employed for the particular boundary condition formulation, e.g. matrix-free iterative solvers.
- Here comes the important concept of dwarf - a mini-application capable of resolving specific task, customized for particular application (as opposed to general-purpose libraries)
- The dwarfs can be used for: weather operations, geophysical research, computer science research and teaching
- This allows for relatively easy interdisciplinary cooperation and teaching, without the burden of NWP framework or multiscale research solver.

COSMO-EULAG dynamical core: dwarf-based structure



Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.
- Written in modern Fortran, capable of both fully static or fully dynamic memory allocation.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.
- Written in modern Fortran, capable of both fully static or fully dynamic memory allocation.
- Using centralized data storage, i.e. no local memory allocation of arrays.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.
- Written in modern Fortran, capable of both fully static or fully dynamic memory allocation.
- Using centralized data storage, i.e. no local memory allocation of arrays.
- Easy to glue together within NWP or research solver, despite being independent of each other.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.
- Written in modern Fortran, capable of both fully static or fully dynamic memory allocation.
- Using centralized data storage, i.e. no local memory allocation of arrays.
- Easy to glue together within NWP or research solver, despite being independent of each other.
- Exposing stencil computations such the stencil is the same everywhere in the computational domain, wherever possible.

Tenets of EULAG dwarfs

- Independent mini-application, able to perform experiments that were published previously or at least verifiable against analytical solution.
- Minimal set of auxiliary code to limit the complexity.
- Capable of parallel runs with 3D MPI decomposition, aiming at (future) MPI+MPI/MPI+OpenACC rather than MPI+OpenMP.
- Written in modern Fortran, capable of both fully static or fully dynamic memory allocation.
- Using centralized data storage, i.e. no local memory allocation of arrays.
- Easy to glue together within NWP or research solver, despite being independent of each other.
- Exposing stencil computations such the stencil is the same everywhere in the computational domain, wherever possible.
- Using halos to compute appropriate boundary conditions (e.g. no flux divergence, zero flux, prescribed flux) by specialized subroutines.

Boundary stencils

In NWP models at the operational core counts, memory bandwidth is often the main performance limiting factor. Therefore it is crucial to reformulate legacy codes to address this issue, especially with regards to the boundary condition implementation. For example, pressure gradient in the x direction is expressed as:

$$px(i,j,k) = (p(i+1,j,k) - p(i-1,j,k))/2dx$$

and the one-sided definition at the border:

$$px(1,j,k) = (p(2,j,k) - p(1,j,k))/dx$$

This can be easily replaced by specifying:

$$p(0,j,k) = -p(2,j,k) + 2 * p(1,j,k)$$

and then executing the central differencing everywhere in the domain as $px_i jk = (p(i+1,j,k) - p(i-1,j,k))/2dx$. Note that now px does not to be stored as a matrix, if e.g. evaluating Laplace operator.

Status of EULAG dwarfs: what is there

- Explicit part of the solution, including upwind, and 2nd order standard and gauge MPDATA schemes. Reference and optimized(towards LAM setup) versions.
- Velocity predictor with linear extrapolation ($itraj = 0$)
- GCR solver with standard preconditioner
- Formulation of linear problem, including outer iteration loop
- Scalar diffusion operator of SGS 2 in legacy and optimized versions
- Vector diffusion operator of SGS 2 in legacy form (ongoing)

What is not there

- Part of periodic boundary conditions
- Spherical option
- Subgridscale models
- Turbulence diagnostics and budgets
- Soundproof options
- 2D option

COSMO-EULAG: dwarf-based dynamical core branch

Timeloop structure:

- Interface to COSMO framework: get first-order forcings and boundary conditions for advection
- Call advection block driver which employs repeatedly MPDATAs from advection dwarf for subsequent fields; the driver is the auxiliary part of the advection dwarf itself, along with the drivers for e.g. idealized tests.
- Provide interpolated boundary conditions in case of subcycling
- Call solver block driver, preparing the linear problem, calling gcr and updating the solution
- Optionally: call EULAG diffusion driver
- Interface to COSMO framework: provide prognostic fields for COSMO parameterizations

Idealized tests

- Advection: rotating sphere, solid body rotation
- Solver: potential flow past the mountain
- Diffusion: 1D and 3D Gaussian

Dwarf: testing implementation of the explicit diffusion operator

Scalar diffusion in physical (e.g. orthogonal: spherical or Cartesian) coordinates is defined in as a divergence of the Fickian flux of the scalar field, e.g. θ'.

$$\mathcal{H} = \frac{1}{\rho}\frac{1}{\partial \bar{x}^j}\left(\alpha\rho\bar{g}^{jk}\frac{\partial\theta'}{\partial\bar{x}^k}\right) \tag{1}$$

Note that proper formulation for momentum is different and it is not yet in the dwarf form/not yet optimized.

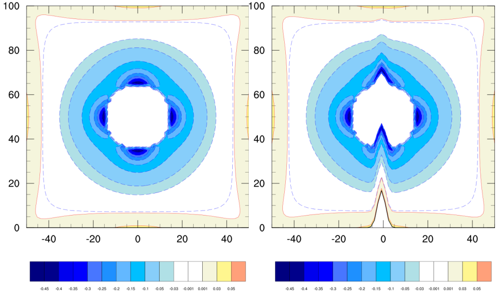
- To test the correctness of the operator formulation in curvilinear coordinates, idealized experiment alluding to the water drop evaporation was employed.

Assuming constant diffusion coefficient K, spherical symmetry and initial-boundary conditions $\psi = 0$ for $r > R$ and $t = 0$, $\psi = \psi_0$ for $r \leq R$ and $t \geq 0$, the diffusion process is described by the analytical solution:

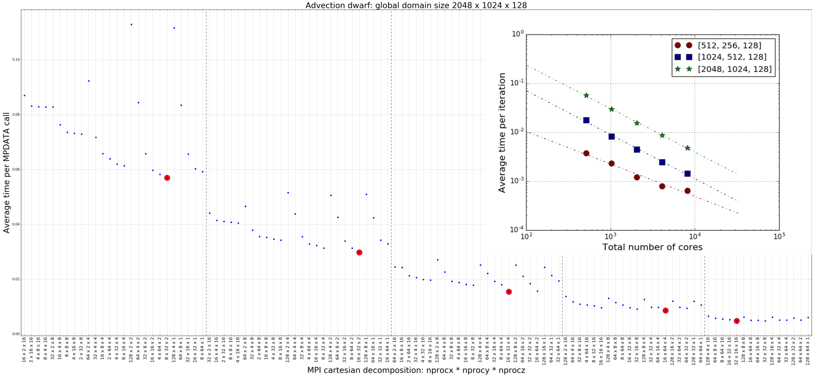
$$\psi(r,t) = \frac{R}{r}\psi_0\left[1 - erf\left(\frac{r-R}{2\sqrt{Kt}}\right)\right]_{r\geq R}. \tag{2}$$

The difference between numerical and analytical solution after is presented in Fig. 1 after the three times of characteristic diffusion time (defined as the time at which the flux of substance ψ at $r = R$, integrated over the sphere's surface, is equal to twice its asymptotic value (at $t = \infty$)). Left panel corresponds to the solution in equidistant Cartesian domain, whereas the right panel corresponds to the solution in presence of Gal-Chen coordinate transformation.

Figure: Diff. between analytical and numerical solution for uniform (left) and curvilinear(right) terrain-following grid



Dwarf: automated performance testing of advection dwarf.



Python script automatically dispatches test program employing MPDATA advection and produces plot presenting optimal 3D MPI decomposition on Cray XC40 at ECMWF. Correspondingly, it produces strong scalability plot.

Subcycling of dynamics

- Dwarf structure facilitates iterative execution of dynamics with smaller timesteps.
- This allows for overcoming the limitations of fully explicit advection.
- Usability limits of such approach are still under investigation, however, preliminary tests against spectral ALARO model of ALLADIN Consortium show that it may allow for integrating conservative COSMO-EULAG with Lagrangian timesteps for physics, at comparable computational costs.

Core count	MPDATA	GCR	Total [s]	Total[kJ]
72	1680	2192	4450	2872
144	1116	982	2287	3413
288	516	617	1326	3228
576	307	347	789	3622
1152	188	201	484	4415

Table: Scalability on Cray XC40 at ECMWF

Conclusions

- Numerics of the established EULAG model was cast in the form of dwarfs and implemented in the operational weather forecasting framework of COSMO Consortium.
- Dwarf structure facilitates evaluation, optimization, hardware-agnostic porting and multidisciplinary development of key weather dynamics solver algorithms.
- COSMO-EULAG solver is now in preoperational testing stage in the Polish weather service; computational efficiency is competitive to the operationally used Runge-Kutta dynamical core.
- Proposed solution addresses efficient use of domain scientists resources and (typically underfunded) HPC dedicated for weather forecasting, as well as builds bridges between single-application COSMO-EULAG implementation targeting weather and multiscale research community EULAG geophysical solver.
- Current and future research directions include integration of multiple equation sets at once, e.g. modeling wind-turbine effects (solver 1) in well developed turbulent boundary layer (solver 2) or integrated Navier-Stokes - Maxwell solvers.