Strategy for implementing a gridpoint solver

Ludovic Auger

ALADIN/HIRLAM Dynamics Day

Motivations : scalability

- In the current Semi-implicit algorithm after variable elimination and projection on vertical modes, 2*Nlev 2D implicit equations are solved :

$$X(x) - \lambda_i^2 \nabla^2 X(x) = rhs(x) \qquad (1)$$

$$X(x) - \gamma^2 \nabla^2 X(x) = rhs(x) \qquad (1')$$

- Equation (1) (1') "Helmholtz type" are solved in spectral space where the solution is trivial.
- Spectral transforms might become more and more computationally expensive, due to global communications on HPC with more and more nodes having their own memory. Current AROME configuration spend less than 10% in Fourier Transform, but it might increase for future configurations.

Motivations : stability

- The reference state for the linearization of the implicit operator does not include orography. With orography linearization leads to a more general operator depending on x, no more projection on vertical mode is possible, coefficients are not constant on x.

$$X(\vec{x},\eta)-G(X(\vec{x},\eta))=rhs(\vec{x},\eta) \qquad (2)$$

$$\text{where } G \text{ is a linear operator} = V(\vec{x},\eta)*\nabla^2$$

- With a grid point solver, system (2) can be solved but not as easily as equation (1).
- It is possible that current instabilities with high slopes might be linked to the implicit system not taking account orography.

METEO FRANCE

Solutions to solve implicit systems in grid point space

- Direct methods are too expensive. With constant coefficients solution is easy to obtain in Fourier space for cartesian coordinates and spherical harmonics for spherical coordinates.
- Simple iterative methods such as Gauss-Seidel, Jacobi, SOR…. are not efficient enough.
- Quasi-Newton methods can be used to solve linear problem but require in general to store and approximation of the Hessian (that is to say an approximation of A), that is too large (although some memory inexpensive version exist).

Solutions to solve implicit systems in grid point space

- The most successful class of method for our problem are Krylov space method ; the solution is seek in the successive Kn vector spaces for a Ax=b system :

$$K_n = \text{span}\{b, Ab, A^2 b \ldots A^{n-1} b\}$$

- Example : Conjugate gradient, Biconjugate gradient, Generalized Minimal Residual...
- Those methods are the most efficient for sparse matrices with a dominant diagonal.
- Among all Krylov space method, Generalized Minimal Residual (GMRES) is the more optimal (meaning that it requires less iterations for a given accuracy).

Solutions to solve the current implicit problem in grid point space

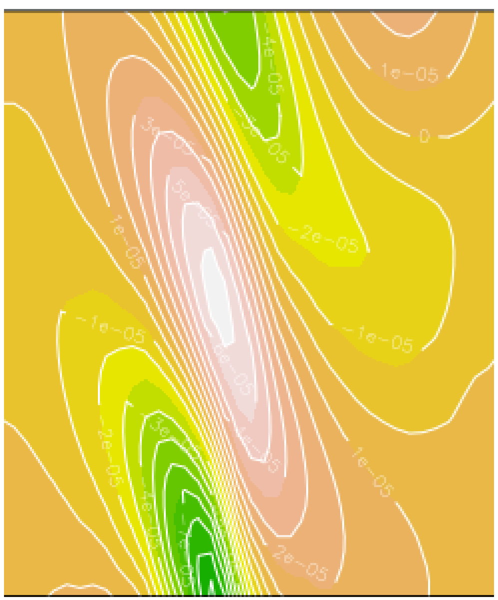
- The current implicit problem leads to 2*Nlev 2D discrete implicit system

$$(I-\lambda_i^2\nabla^2)d_i^{n+1}=d^*$$
$$\text{for } i=1\ldots N_{lev}$$
$$\text{then}$$
$$(I-\gamma^2\nabla^2)D_i^{n+1}=D^*$$
$$\text{for } i=1\ldots N_{lev}$$

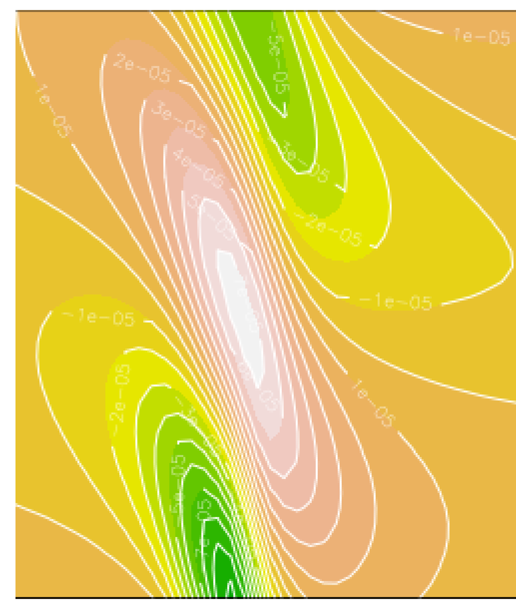
- Operators $(I-\lambda_i^2\nabla^2)$ and $(I-\gamma^2\nabla^2)$ are $(Nx*Ny)^2$ sparse matrices that depends on the vertical. In matrix writing on a given line the number of non-zero coefficient depends on the choice of the order for the derivative operator (9 non-zero coefficients for a 4th order derivative).
- For vertical divergence, most of the vertical modes are solved with a very small number of iterations.

Example in 2D AROME model

- Test were performed with the iterative solver GMRES and the same variable elimination + projection into vertical modes
- Test with hydrostatic orography (5km length, 200m height)
- dx=2000m, dt=60s, predictor-corrector. The iterative solver uses 16 iterations at each time step, that is to say one iteration every 4s



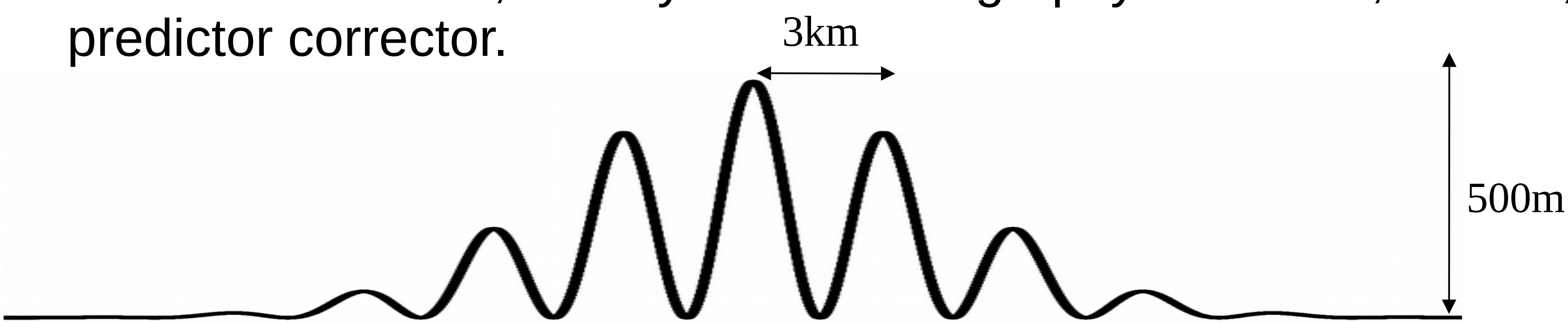
experiment



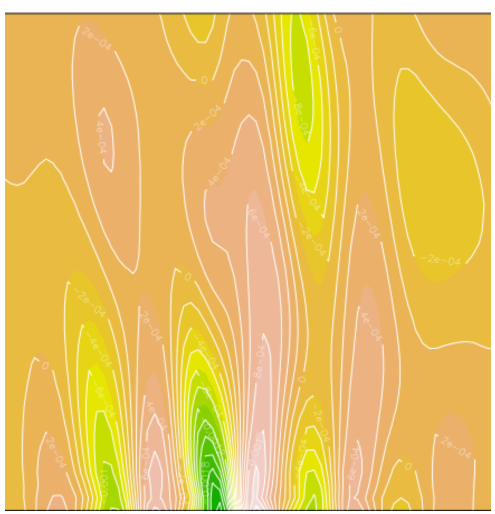
reference

Example in 2D AROME model

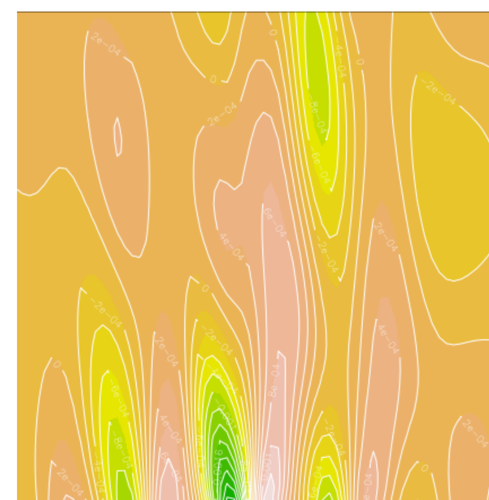
- More difficult case, non-hydrostatic orography dx=400m, dt=20s, predictor corrector.



- Iterative scheme uses around 35 iterations, that is to say one iteration every 0.5 s of forecast time.



experiment



reference

Positive points

- The condition number is ~ Courant number^2 that is to say ~100. That corresponds to quite well conditionned problems.
- In term of communication, roughly, the frequency and amount of data exchanged for a given forecast lead time seem to be equivalent compared to a HEVI model.
- If results are confirmed in 3D, that could be the first step for moving from a spectral model to a full grid-point model.

Negative points

- The number of iterations can depend on the meteorological situation (not very convenient for operations).
- Memory cost can be high for GMRES, other algorithms (congrad) with low recursivity require a few more iterations.
- Accuracy is only evaluated as a residual: we do not now $|x-x_{true}|$ but rather $|Ax-Ax_{true}|$.
- Necessity to test in real 3D model.

Further testing/implementation

- Testing in AROME/ALADIN dynamical core could be done as follows :
- In routine **ESPNHSI** :
- Insert a spectral to gridpoint transform
- Call the solver. The solver will call an horizontal gridpoint derivative computation routine. The solver will perform the minimization for every vertical mode at the same time
- Then call gridpoint to spectral transform
- Perform the same for horizontal divergence.

Further testing/implementation

- First part of ESPNHSI

ESPNHSI

- **SI_CCOR**
- **SISEVE**
- **SITUU**

Computes rhs for d

MXMAOP Vertical mode transform

“Helmholtz” inversion

MXMAOP Vertical mode backward transform

Further testing/implementation

- First part of ESPNHSI

ESPNHSI

SI_CCOR
SISEVE
SITUU
} Computes rhs for d

MXMAOP Vertical mode transform

Spectral transform

PDG solver → **Outer loop**
Inner loop
Gridpoint derivatives (requires communication with neighbours)

Backward transform

MXMAOP Vertical mode backward transform

Further testing/implementation

- Different solvers can be tested with that implementation.
- That implementation might be relevant for testing, but it is also the first step for a full gridpoint code.
- Spectral diffusion also requires to be performed in gridpoint space.

Conclusions/Discussion

- Testing in 2D context is promising, but it is difficult to anticipate the result in full 3D context
- Convergence theorem for gridpoint solvers shows that the number of iterations is function of Courant number^2, since Courant number should not incease for future resolutions consequently scalability could be preserved.
- Steep slopes issues might be more problematic than scalability, if we are not able go beyond 500m resolution.
- Should we start implementation in common code.