Vertical finite element discretisation in NH kernel of model system AAA

Jozef Vivoda, Prague, 05-06/2015

December 11, 2015

1 Introduction

We have performed following activities during 6 weeks period

1. coding of back-compatible version of VFE scheme

   During the last years the divergence has occurred in between VFE scheme version presented during workshop in Island and the scheme coded in our latest version of development code. We have introduced new key *LVFE_COMPATIBILITY* to ensure that our latest development code is backward compatible with VFE scheme version, we used to perform very high resolution 3D real tests.

   The key *LVFE_COMPATIBILITY* affects subroutines

   - gnh_tndlagadiab_gw.F90
     BCs treatment and formulation for term $\frac{\partial(p-\pi)}{\partial\eta}$ changed.
   - gnhgw2svd.F90
     Derivative $\frac{\partial gw}{\partial\eta}$ replaced by derivative $\frac{\partial gw-gw_s}{\partial\eta}$ and BCs changed accordingly.
   - gnhpreh.F90
     Treatment of BCs in term $\frac{\partial\hat{q}}{\partial\eta}$ has been changed.
   - siseve.F90
     BCs for terms $\frac{\partial\hat{q}}{\partial\eta}$, $\frac{\partial^2\hat{q}}{\partial\eta^2}$, $\frac{\partial\frac{\pi^{*2}}{m}}{\partial\eta}$.
   - suvertfe.F90
     Change of BC for operators RINTE, RINTBF11, RDERBF01 and RDDERBF01.
   - suvertfeb.F90
     Change of boundary properties of test functions w used in weighted residual method (see Appendix A).

2. cleaning of code

   We cleaned the code of obsolete keys. Consolidation of existing developments keys was done.

3. work on article to be posted to MWR

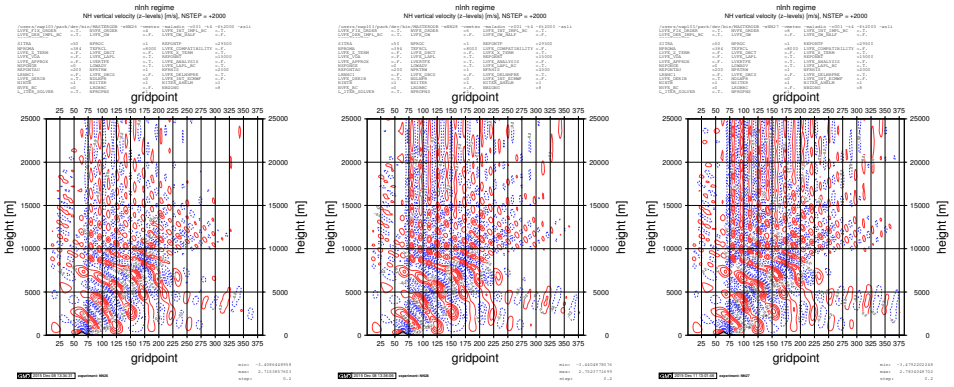
   We have ensured compatibility of text in article draft with model code we use to run tests that will be published.

4. bugfixing order of splines

   We knew that there was bug in our code that allow us to run VFE scheme only with splines of order 4. Analysis of discretisation accuracy was made in the past for diferent orders of B-splines. It was shown that increasing order of spline improves overall accuracy of the scheme. We have tested this aspect of the scheme.

   We tested the scheme with fixed order of splines and implicit boundary conditions of VFE operators (see Appendix A for detailed explanation).

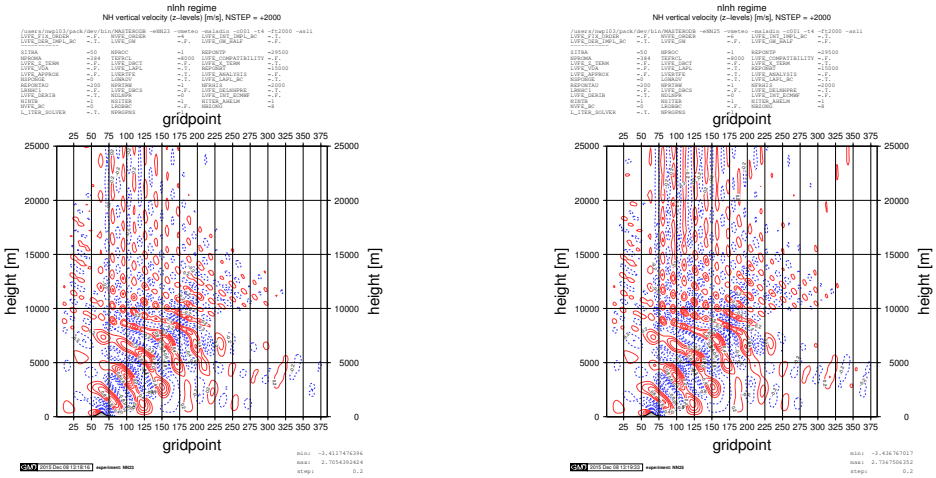
   The results from NLNH test case tested with order of splines 4, 6 and 8 with fixed order ($LVFE\_FIX\_ORDER = TRUE$) are shown on Figure 1. Also the experiments with odd order of splines were unstable. This must be further investigates. We did not identify yet, if the instability in this case, is property of the scheme, or just remaining bug in the code.



(a) $NVFE\_ORDER = 4$ (b) $NVFE\_ORDER = 6$ (c) $NVFE\_ORDER = 8$

Figure 1: NLNH flow regime - test of spline order with fixed order of spline (a) 4th order cubic splines used, (b) 6th order splines used and (c) 8th order splines used.

   The results with order of splines 4 and 6 with fixed number of knots ($LVFE\_FIX\_ORDER = FALSE$) are shown on Figure 2. The experiment of order of spline 8 was unstable. The odd order of splines was unstable again as in the previous case.



(a) $NVFE\_ORDER = 4$ (b) $NVFE\_ORDER = 6$

Figure 2: NLNH flow regime - test of spline order with fixed knots to contruct basis functions. In this case the order is reference order for operator with no boundary conditions (more in text) (a) 4th order cubic splines used, (b) 6th order splines used.

5. gw on full levels with VFE treatment vs. gw on half levels with FD treatment

   $LGWADV$ key activates grid point computations with gw as a prognostic quantity.

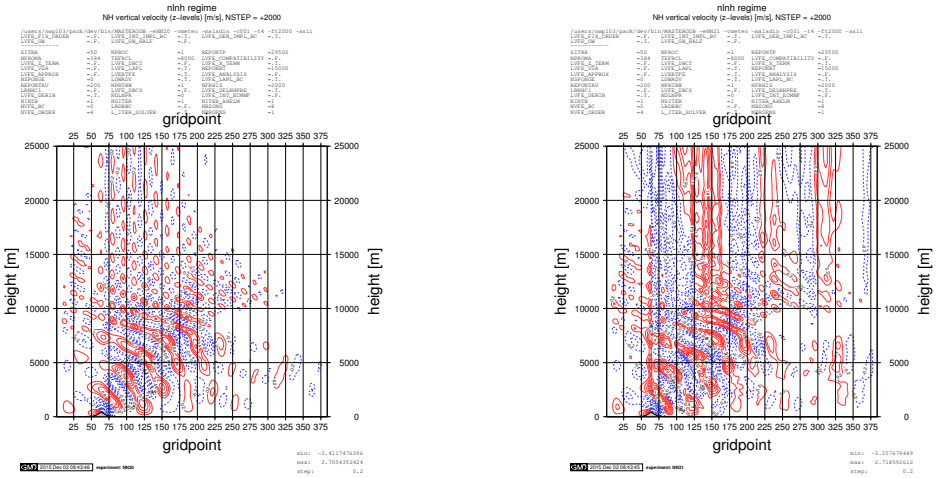
   There are 3 possible approaches to treatment of this quantity controlled by keys $LVFE\_GW$ and $LVFE\_GW\_HALF$. Meaning of namelist parameters can be understood from following table

scheme	$LVFE\_GW$	$LVFE\_GW\_HALF$	test
FD scheme with gw on half levels	F	F	NN20
VFE scheme with gw on full levels	T	F	NN21
VFE scheme with gw on half levels	F	R	NN22

   The transformation from d_4 to gw and back must be the identity transform. This ensures correct steady state solution in the case of atmosphere in hydrostatic balance (no dynamical tendency to gw). FD derivator and integral operators fullfill this condition in test NN20.

   We re-introduce the invertible derivative and integral VFE operators acting on full levels in test NN21 (invertible operators are defined under subroutine SUVERTFE as $RDERGW$ and $RINTGW$).

   NLNH experiment was used to test this possibility. The result is compared against FD scheme with gw on half levels on Figure 3.



(a) gw on half levels

(b) gw on full levels

Figure 3: NLNH flow regime test with gw tranformations to d variables treated in FD manner (a) and in VFE manner (b). In (a) gw is defined on half levels and in the case of (b) on full levels. VFE invertible operators were used in (b) and there is apparent problem with noise.

We started implementation of key *LVFE_GW_HALF*

6. further tests of fixed order vs. fixed knots options

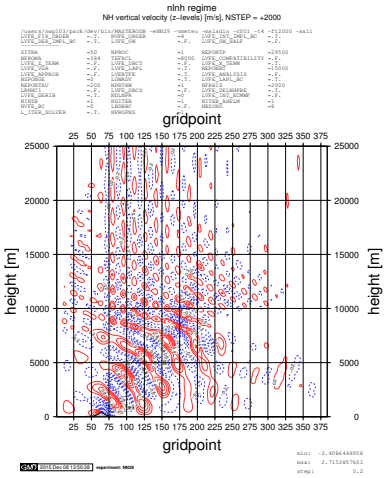
   Fixed knots are mandatory in order to formulate invertible operators. We have continue in investigation of this option with bug-fixed code. The results are already shown on figure 2 and 1. There is apparent difference between the choices. We are not able to conclude so far which approach is more appropriate as the differences with 2D tests were stable and very similar. We will continue in 3D real cases.

7. operators with explicit vs. operators with implicit boundary conditions

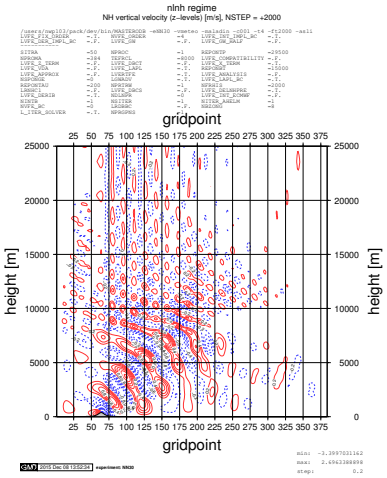
   We implemented the keys to control the spline fit of input function. The key *LVFE_DER_IMPL_BC* control the treatment of boundary conditions and derivative operators and the key *LVFE_INT_IMPL_BC* in integral operators.

   The same experiment (NLNH) was used to test this option. We see the effect of

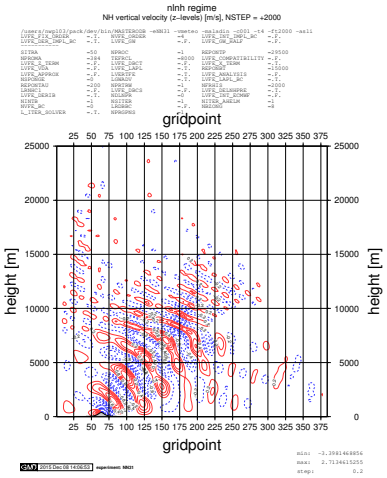
   Comment:
   We found sensitivity to this operators for *LVFE_GW* (gw scheme with VFE operators and gw on full levels) and $NITER\_HELM > 1$). The scheme become unstable and it is still opened issue to investigate this instability.



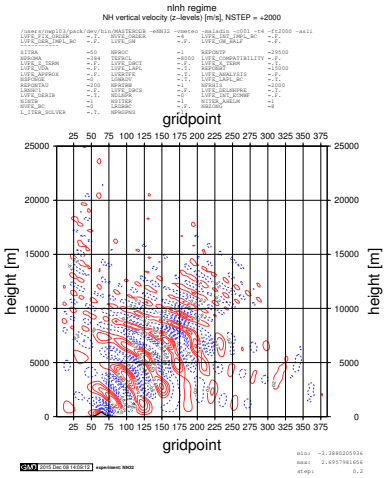
(a) $LVFE\_DER\_IMPL\_BC = T$ and $LVFE\_INT\_IMPL\_BC = T$



(b) $LVFE\_DER\_IMPL\_BC = F$ and $LVFE\_INT\_IMPL\_BC = T$



(c) $LVFE\_DER\_IMPL\_BC = T$ and $LVFE\_INT\_IMPL\_BC = F$



(d) $LVFE\_DER\_IMPL\_BC = F$ and $LVFE\_INT\_IMPL\_BC = F$

Figure 4: NLNH flow regime test. Explicit vs. implicit boundary conditions used in operators

2 Appendix A

We discretize operator g applied on continuous function $f(\eta)$. Operator g can be integral, derivative or second derivative operator and we can write

$$g\left(f(\eta)\right) = d(\eta) \quad . \tag{1}$$

We expand

$$f(\eta) = \sum_{i=1}^{L} \hat{f}_i B_i(\eta) \tag{2}$$

and

$$d(\eta) = \sum_{i=1}^{L} \hat{d}_i D_i(\eta) \tag{3}$$

using two independent set of basis functions B_i and D_i.

Having $f(\eta)$ sampled on L model levels, we can transform it from physical to VFE space as $f(\eta_k) = \sum_{i=1}^{L} \hat{f}_i B_i(\eta_k)$. The same holds for $d(\eta)$.

When we substitute these expansions into differential form we obtain

$$\sum_{\mathbf{i=1}}^{\mathbf{L}} \hat{\mathbf{f}}_\mathbf{i} \mathbf{g}\left(\mathbf{B_i}(\eta)\right) = \sum_{\mathbf{i=1}}^{\mathbf{L}} \hat{\mathbf{d}}_\mathbf{i} \mathbf{D_i}(\eta). \tag{4}$$

Weak form of differential form is obtained with arbitrary set of L weighting functions w

$$\sum_{i=0}^{L+1} \left[\int_{\mathbf{0}}^{\mathbf{1}} \mathbf{g}\left(\mathbf{B_i}(\eta)\right) \mathbf{w_j}(\eta) \mathbf{d}\eta \right] \hat{f}_i = \sum_{i=0}^{L+1} \left[\int_{\mathbf{0}}^{\mathbf{1}} \mathbf{D_i}(\eta) \mathbf{w_j}(\eta) \mathbf{d}\eta \right] \hat{d}_i \qquad j = 1, \ldots, L. \tag{5}$$

Spline basis functions B, D and w are independent.

Properties of VFE operators are controlled by following keys:

1. *LVFE_FIX_ORDER*

   As mentioned in previous reports the number of basis function needed to cover whole domain with correct BCs is $N_b = NFLEVG + BCs$. The number of knots needed to construct such basis is $N_{knots} = N_b + C$. So we could fix either order of splines C when we define $LVFE\_FIX\_ORDER = TRUE$ or we can fix the number of knots N_{knots} and to adjust the order of spline ($LVFE\_FIX\_ORDER = FALSE$). In this case the order of spline defined via *NVFE_ORDER* is valid for basis with no BCs.

   subroutines: VFE_OPER_SETUP

2. *NVFE_ORDER*

   Reference order of B-splines C. In the case of $LVFE\_FIX\_ORDER = FALSE$ it is the order of B splines constructed with no BCs considered.

3. *LVFE_INT_IMPL_BC*

   Implicit treatment of boundary conditions in integral operator. Detailed explanation under *LVFE_DER_IMPL_BC*.

4. *LVFE_DER_IMPL_BC*

Implicit treatment of boundary conditions in derivative operators (first and second derivative as well).

The transformaton between physical space and VFE space (2) is defined in matrix form as

$$\hat{f}_i = T_{ik}^{-1} f_k \tag{6}$$

with $T_{ik} = B_i(\eta_k)$. The spline fit is then $f(\eta) = \hat{f}_i B_i(\eta)$. If we require some specific condition to be fulfilled at boundary condition we have two possibilities how to do it

- $LVFE\_DER\_IMPL\_BC = TRUE$ - **implicit treatment of BCs**
  we enforce our basis function to fullfill required BCs. In this case the dimension of transformation matrix is $NFLEVG$.
  Example:
  $f(\eta)$ is sampled at L full levels with one BC $\frac{\partial f(0)}{\partial \eta} = 0$ at the model top and $f(1) = 0$ at model surface. We contruct basis functions with property $\frac{\partial B_i(0)}{\partial \eta} = 0$ and $B_i(1) = 0$. This is done in a way that we construct $L+2$ basis functions using De Boor's algorithm (see previous report for more information) . We ensure property at model top by linear combination of first two basis functions $B_2 = B_1 + B_2$ and omitting B_1 afterwards. The condition at model surface is fulfilled by all basis function except B_{L+2}, therefore this function is not used. We use then L basis functions only (we omitted first one and the last basis functions from our set of $L+2$ basis functions).

  subroutine: *vfe_oper_implicit_bc*
- $LVFE\_DER\_IMPL\_BC = FALSE$ - **explicit treatment of BCs**
  we add required BCs into vector f_k. In that case the dimension of transformation matrix T_{ik}^{-1} is $NFLEVG + BCs$.
  Example:
  $f(\eta)$ is sampled at L full levels with one BC $\frac{\partial f(0)}{\partial \eta} = 0$ at the model top and $f(1) = 0$ at model surface.. The transformation will be

$$\begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \dots \\ \hat{f}_L \\ \hat{f}_{L+1} \\ \hat{f}_{L+2} \end{bmatrix} = \begin{bmatrix} \frac{\partial B_1(0)}{\partial \eta} & B_1(\eta_1) & B_1(\eta_2) & \dots & B_1(\eta_L) & B_1(1) \\ \frac{\partial B_2(0)}{\partial \eta} & B_2(\eta_1) & B_2(\eta_2) & \dots & B_2(\eta_L) & B_2(1) \\ \dots & & & \dots & \dots & \\ \frac{\partial B_{L+1}(0)}{\partial \eta} & B_{L+1}(\eta_1) & B_{L+1}(\eta_2) & \dots & B_{L+1}(\eta_L) & B_{L+1}(1) \\ \frac{\partial B_{L+2}(0)}{\partial \eta} & B_{L+2}(\eta_1) & B_{L+2}(\eta_2) & \dots & B_{L+2}(\eta_L) & B_{L+2}(1) \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial f(0)}{\partial \eta} = 0 \\ f(\eta_1) \\ \dots \\ f(\eta_{L-1}) \\ f(\eta_L) \\ f(1) = 0 \end{bmatrix}$$

Here we use $L+2$ general functions defined according De Boor (more information can be found https://en.wikipedia.org/wiki/De_Boor's_algorithm).

This two approaches seems to be equivalent at the first glance, but they are not. This is shown on the following figure where the derivative of $f(\eta) = (sin(\pi\eta)^2 cos(\pi\eta))$ is shown. This results is for $NFLEVG = 10$ and boundary conditions $\frac{\partial f(0)}{\partial \eta} = 0$ and $f(1) = 0$. What we see the error of derivative computed with operators *RDERBF10* resp. *RDERBF10_IMPL* on figure 5. The error is apparently different.

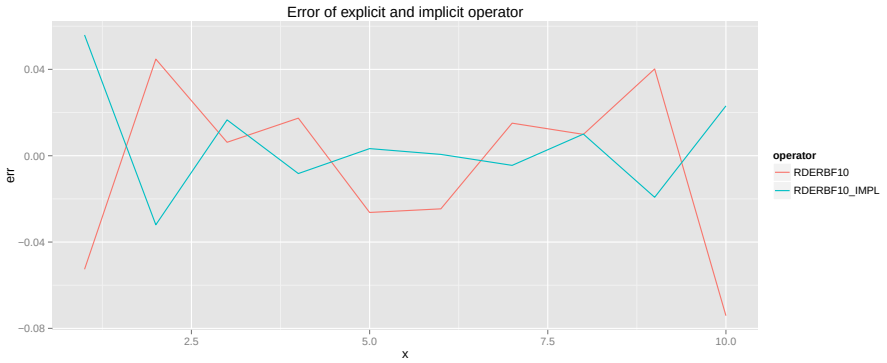


Figure 5: Error of derivative operator with explicit and implicit BCs incorporated into transformation from physical to VFE space. Testing function was $f(\eta) = (sin(\pi\eta)^2 cos(\pi\eta))$ sampled at 10 full levels with BCs $\frac{\partial f(0)}{\partial \eta} = 0$ and $f(1) = 0$. The derivative error of operator with explicit conditions is plotted with red line and with implicit with blue line. Horizontal axis represents index of full level where derivatives were computed.