go to poblano folder and run install_poblano.m
go to MORTestbed_UTEP/Nonlinear/ and run init.m

Workflow0 runs original ROM and GNAT
Workflow1 runs constrained ROM, GNAT using constraints from ROM (reconstructed full vectors)
Workflow2 runs constrained ROM with approximated constraints in GNAT (there is a bug in the code; jacobian of the right flux and sum of force terms are incorrect. Don't know how to fix that).
Workflow3 change number of cells and run constrained ROM on multiple domains and GNAT with reconstructed constraints(from ROM):

3*ncell < number of basis vectrors: LSPG_subdomains()
3*ncell == number of basis vectrors: solveConstraints()
3*ncell > number of basis vectrors: minimizeConstraints()

Workflow4 change number of cells and run constrained ROM on multiple domains and GNAT with approximated constraints(from ROM)

To debug Workflow2
Run **Workflow2** → solves FOM→ ROM

Workflow2 chooses constrained method and calls function **ConstraintsOneDomain_old**
(norm of constraints are 0 after each timestep, also checked with poblano -- derivative is right (can uncomment to check it again)

there is another function that computes constraints- **constraintsForGNAT**. It is used in GNAT. In i**mprovedSLPG** it is checked to be right. Can uncomment if want to check again.

at the end of ROM build basis for left, right fluxes and for force term. For debugging purposes on the very first step save DforceQ, romdroeF, J2L_true, J2R_true (left, right jacobians), force term Q2_true. Also saved romroeF, Fright, Fleft, forceQ and built basis without truncation.

In GNAT.m file debug the approximated constraints

% constraints are build using
% governEqGNAT, ResJacGNAT, ... from quasi1dEuler.m (Nonlinear/Classes/Problem)
% TimeIntNLFuncGNAT from quasi1DImplicitPseudoTime (Nonlinear/Classes/Time)
% computeJPHIhat from GNAT.m

solver==3 → call **GnatConstraints**()

for debugging use poblano

check right, left fluxes and force term with derivatives (uncomment if need to check again)- it shows that computed right

also **ApproxConstr** function is checked with poblano, it's right

HOWEVER, if we load the true right, left fluxes with their Jacobians on the first step as well as force term with it's Jacobian, we get the following:
(reconstruct the full right, left fluxes and load true fluxes saved when running Rom. Compute the norm of the difference save; for calculating jacobian ; uncomment if want to recheck)

myLeftFlux3fast- Left Flux and its Jacobian are correct (uncomment to load saved data and check)

myRightFlux3fast- Right Flux is correct, but Jacobian is wrong (uncomment to check)

myQdQfast - incorrect reconstruction of Q, sq and dsq

CODE CAN BE FASTER, FOR EXAMPLE IF RIGHT AND LEFT FLUXES ARE CALCULATED IN THE SAME FUNCTION. I SPLITED THEM SINCE IT WAS EASIER TO DEBUG. ALSO IN ROM FILE CONSTRAINTS CAN BE COMPUTED ONLY USING CONSTRAINTS FOR GNAT FUNCTION. I DIDN'T CHANGE IT FOR DEBUGGING PURPOSES.