Naval Research Laboratory
Washington, DC 20375-5320

NRL/6003/MR—2024/1

Configuration Timings for the HYCOM Gulf of Mexico Scenario

JAY BORIS

Chief Scientist
Material Science and Component Technology Directorate

KEITH S. OBENSCHAIN
YU Y. KHINE

Laboratory for Advanced Computational Physics Branch
Laboratories for Computational Physics and Fluid Dynamics Division

ROBERT O. ROSENBERG

Scientific Applications of High Performance Computers (HPC) Section
Information Technology Division

CLARK D. ROWLEY

Ocean Data Assimilation and Probabilistic Prediction Section
Ocean Sciences Division

TOMMY G. JENSEN

Nearshore and Coupled Model Systems Section
Information Technology Division

PRASAD G. THOPPIL

Open Ocean Processes and Predictions Section
Ocean Sciences Division

GOPAL PATNAIK

Alion Science and Technology Corporation
McLean, VA

March 5, 2024

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
03-05-2024	NRL Memorandum Report	10-01-2022	01-24-2024

4. TITLE AND SUBTITLE

Configuration Timings for the HYCOM Gulf of Mexico Scenario

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		1P79

6. AUTHOR(S)

Jay P. Boris, Keith S. Obenschain, Yu Y. Khine, Robert Rosenberg, Clark Rowley, Tommy G. Jensen, Prasad G. Thoppil, and Gopal Patnaik*

7. PERFORMING ORGANIZATION / AFFILIATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Ave SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/6003/MR—2024/1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR / MONITOR'S ACRONYM(S) NUMBER	11. SPONSOR / MONITOR'S REPORT NUMBER(S)
Naval Research Laboratory 4555 Overlook Ave SW Washington, DC 20375-5320	NRL	

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTAL NOTES

This is the first NRL Memorandum Report in a possible series describing some of our work to improve the performance of the Navy's open source oceanography code HYCOM. R&D on use of reduced precision variables to reduce data transport and on hybrid parallelization techniques is reported. An effective optimization strategy is presented.

*Alion Science and Technology Corp., 8350 Broad Street #1400, McLean, VA 212102

14. ABSTRACT

See Report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	c. THIS PAGE		
U	U	U	SAR	19

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jay P. Boris	(202) 767-3055

This page intentionally left blank.

Configuration Timings for the HYCOM Gulf of Mexico Scenario

Abstract

HYCOM, standing for HYbrid Coordinate Ocean Model, is a general circulation model for the oceans (see Large, et al., 1994; Bleck, 2002; Wallcraft, et al., 2009). The primitive fluid dynamics equations are solved for the horizontal components of the flow and a reduced-order K-Profile-Parameterization (KPP) model is solved in the vertical to treat convection, diffusion and turbulent mixing implemented as a calibrated implicit diffusion representation. The hybrid grid in HYCOM adapts to treat deep ocean and shallow coastal regions appropriately. The model includes surface driving terms, ice and salinity physics and treats all the differentiated ocean layers.

As part of a project to study reduced precision computation and other computational approaches to accelerate computing HYCOM regional solution ensembles, this report describes the use of 'Diablo' interactive graphics embedded in HYCOM to execute an extended campaign of HYCOM Gulf of Mexico simulations to time various parallel processing configurations that involve one or more HPC computing nodes and different configurations of Open MP parallelism within the Message Passing Interface parallel framework built into HYCOM. The metric for comparing the different parallel approaches is the 'core efficiency,' that is 'bang for the buck,' using the NRL 32-node system 'luigi' where each node has 128 cores and 256 gigabytes of memory but where all 128 cores are not used in many of the configurations tested. The optimal 'core efficiencies' are found when a single node is filled with a number of MPI tasks (ranks) that leaves enough cores available for effective OpenMP parallelism within each task.

This page intentionally left blank.

1. Introduction and Summary:

HYCOM: "The HYbrid Coordinate Ocean Model (Large, et al., 1994; Bleck, 2002; Wallcraft, et al., 2009) is a primitive equation, general circulation model. The vertical coordinates are isopycnal in the open, stratified ocean, but use the layered continuity equation to make a smooth transition to terrain-following coordinates in shallow coastal regions, and to Z-level coordinates in the mixed layer and/or unstratified seas. The hybrid coordinates extend the geographic range of applicability of traditional isopycnic coordinate circulation models toward shallow coastal seas and unstratified parts of the world ocean. HYCOM maintains the significant advantages of an isopycnal model in stratified regions while allowing more vertical resolution near the surface and in shallow coastal areas, hence providing a better representation of the upper ocean physics. HYCOM provides a major advance over previously existing operational global ocean prediction systems, since it overcomes design limitations of those systems as well as limitations in vertical and horizontal resolution. The result appears as a more streamlined system with improved performance and an extended range of applicability (e.g., other systems are seriously limited in shallow water and in handling the transition from deep to shallow water).

The Gulf of Mexico Simulation Campaign: The Gulf of Mexico scenario provides a realistic case with roughly square 4 km cells that starts from a realistic multi-day initial condition with all physical forcing functions and boundary conditions (etc.) and runs up to 30 days. Seeking relative timings of nominally identical physical cases using different parallel implementations, all the simulations in this campaign were 2 days in real time duration with run times generally between 3 and 15 minutes. The grid is 541 cells west to east (x) and 385 cell south to north (y) with 36 hybrid-grid z levels (depth). Typical results will be shown in following sections using an interactive graphics capability. We then discuss an extensive configuration timing table and its relevance to the goal of increasing the number of ensemble members available for operational forecasts.

This Gulf of Mexico test problem is being used to study reduced precision and other computational approaches to accelerate computing regional ensembles of oceanographic solutions. Ensembles of similar simulations are used to test the range of probable results where input conditions, forcing functions, and atmospheric variability are uncertain and thus users need to know the likely range of future forecast results. Each ensemble member is a separate simulation for the forecast duration so the overall ensemble problem is 'embarrassingly' parallel. Not only can Message Passing Interface (MPI) and Open Multi-Processing (OpenMP) approaches be used together in many current high-performance computer architectures implementing fluid dynamic scenarios, they compete for resources and thus their use must be balanced for optimal performance.

Such is the case with HYCOM. An optimal implementation will be hardware, software, and grid architecture dependent. The metric for comparing the different parallelization and grid configuration approaches used here is the overall 'core efficiency,' that is 'bang for the buck.' The cost of the computation on a single core is the normalizing factor. 'Core efficiency' is the total time of a parallel HYCOM benchmark run divided by the run time of a reference single-core run, which is then divided by the total number of cores reserved. The run time is how long do the computations take after initialization? This campaign was performed on NRL's 32-node 'luigi' AMD EPYC system where each node has 128 cores and 256 gigabytes of memory. We find that the practical maximum core efficiency is about 25% and is found when a single 'luigi' node with 128 cores is reserved for each ensemble member simulation.

Manuscript approved February 7, 2024.

2. Diablo:

This report also introduces the use of the 'Diablo' interactive graphics package, now embedded in HYCOM, and applied to the extensive campaign of HYCOM Gulf of Mexico simulations reported here. Diablo is a 2D visualization and graphing tool for modeling and simulation (m&s) software. Labelling can be added to the plots in Diablo and simple 3D plots can be constructed on the fly. Diablo was developed as a diagnostic tool to allow modeling and simulation developers and users to interact with a simulation via a graphical user interface, to look at data visually, to pause and step through a simulation, and to change variables that impact the running of the simulation. It is tightly coupled with the running simulation and has been integrated with both MPI and OpenMP based parallel codes. Diablo has a minimum impact on simulation running time. Embedding the graphics costs only a couple of percent for a number of frequent plots and provides instant feedback on simulation progress, numerical problems, instabilities, and correctness. The primary purpose of this simulation campaign and report is to consider tradeoffs toward an optimal way for HYCOM to calculate a forecast ensemble of related simulations.

Diablo runs through a wrapper around the time-stepping physical and fluid dynamic simulations while allowing remote interaction with the ongoing simulation. Diablo implements QT and uses its window control libraries and protocols. In practice this is much like the master of ceremonies/host on a T.V. game show. At user-selected timesteps in a run and possibly specific locations in the domain a user-routine is called to prepare the desired plots. These plots are constructed pixel-by-pixel using a user-provided color map. With a call to a window control routine, Diablo converts the 2D array of color-mapped pixels to a PNG file and 'shepherds' that file's transfer from the large host computer to a remote laptop or workstation. The results of multiple files are thus instantly displayed on distinct local windows. A call to another Diablo routine writes selected PNG files to designated directories on the host computer. Most of this is asynchronous so the simulation can resume whenever the graph preparation is complete, as Diablo is transferring and displaying the PNG images remotely.

The following four figures were each composited from 6 Diablo plots of four of the HYCOM primary variables at three different z levels in the simulations. These z levels vary in depth from place to place and with time during the simulations so only relative depths between $k_z = 1$, $k_z = 12$, and $k_z = 20$ are involved. The three HYCOM simulations illustrated in these four figures were all performed on 21 August 2023. Figures 1 and 2 show the "initial" conditions for the 2-day runs, which is actually the result of a 44-day spin up simulation performed to establish all the input functions and files for this test case.

In the examples below, the overall 2D domain is masked to identify the ground as grey using the data available in HYCOM. The 541 by 385 cell domain in the total grid is shown in the figures but only about 75% is water. In HYCOM the different variables are defined on different interlaced grids so you may be able to spot slightly different renditions of the shoreline between the two velocity components and the scalar variables salinity and temperature. The five primary variables where each scanned over the two days for the maximum and minimum values found in water cells at z level 12 in the grid. The fifth variable potential density 'th3d' is not plotted in the figures. These fixed maximum and minimum values are constant in time and are used to set the limits of the color maps in each figure. Thus the same color on different plots of a particular variable indicate the same physical value regardless of location on the plot or the time. Simple routines are available for labelling the individual PNG images, as shown.

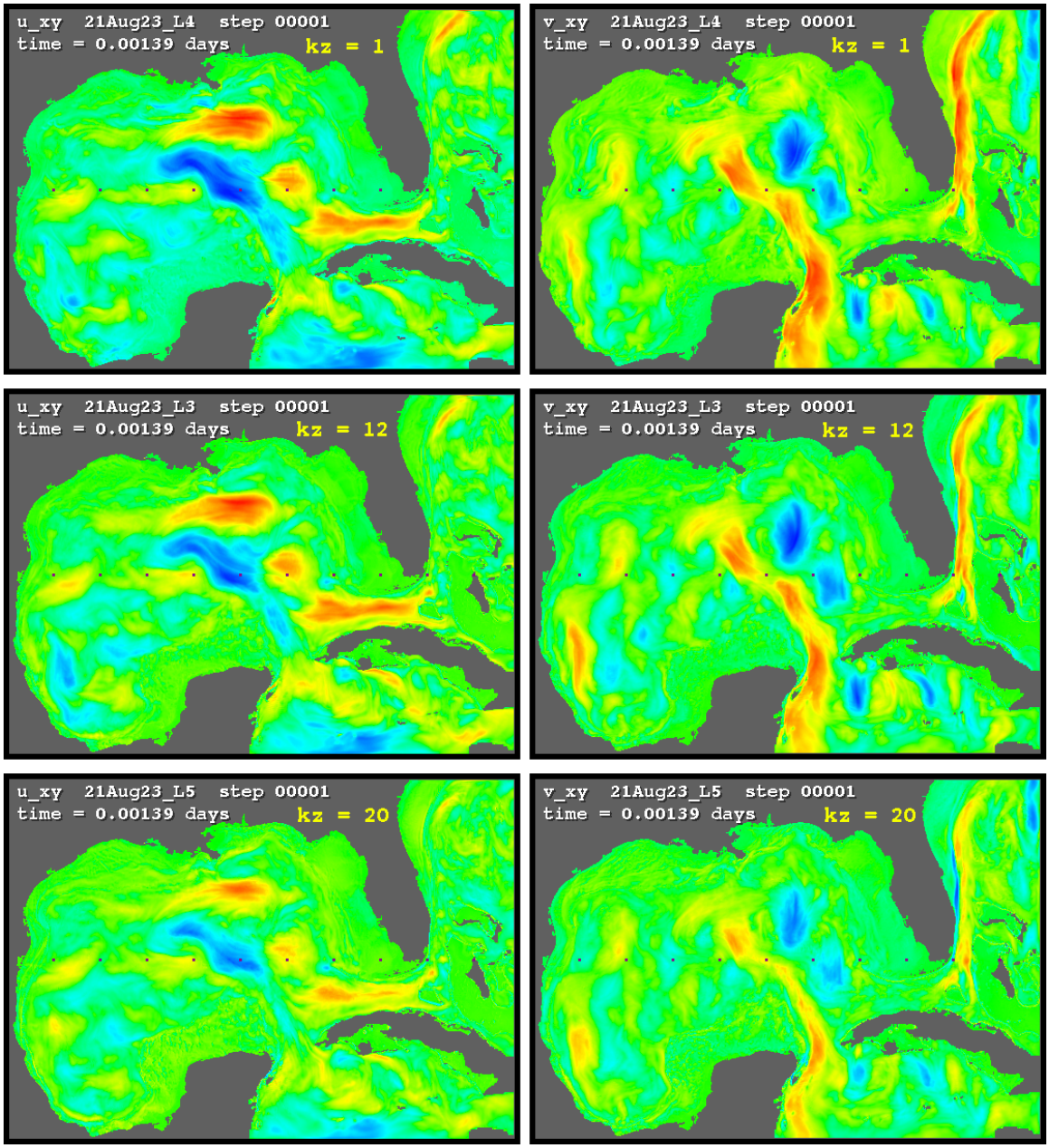

Fig 1. Initial horizontal flow velocities at 3 different z levels in the 2-day Gulf of Mexico HYCOM simulation (timestep 1). Left panels (top to bottom): East-west flow velocity u_xy (m/s) at three levels k_z = 1, 12, and 20 of the 36 gridded levels. Right panels (top to bottom): North-south flow velocity v_xy (m/s) at the same three levels. The color map is a spectrum from dark blue, -1.45 m/s, to bright red, +1.45 m/s. The 10 magenta squares along the mid line of the Gulf are stations for future vertical grid layer thickness printouts. File Fig 1. PT_step1_3Z_u_xy&v_xy.png

Figure 2 shows the east-west and north-south velocity components at the three z grid levels after two days have elapsed following 1440 2-minute timesteps. Two days is long enough to get reasonably accurate timings of the various MPI-OpenMP parallelization combinations, but the runs were short enough, several minutes of wall time, to allow many runs in a day. After only two days run time, however, the changes in the solution from the fully evolved starting conditions in Figs. 1 and 3 (below) are visible but still small. In Figs. 1 and 2 the horizonal velocity components are stronger near the surface and clearly conform to the various nearby land masses.

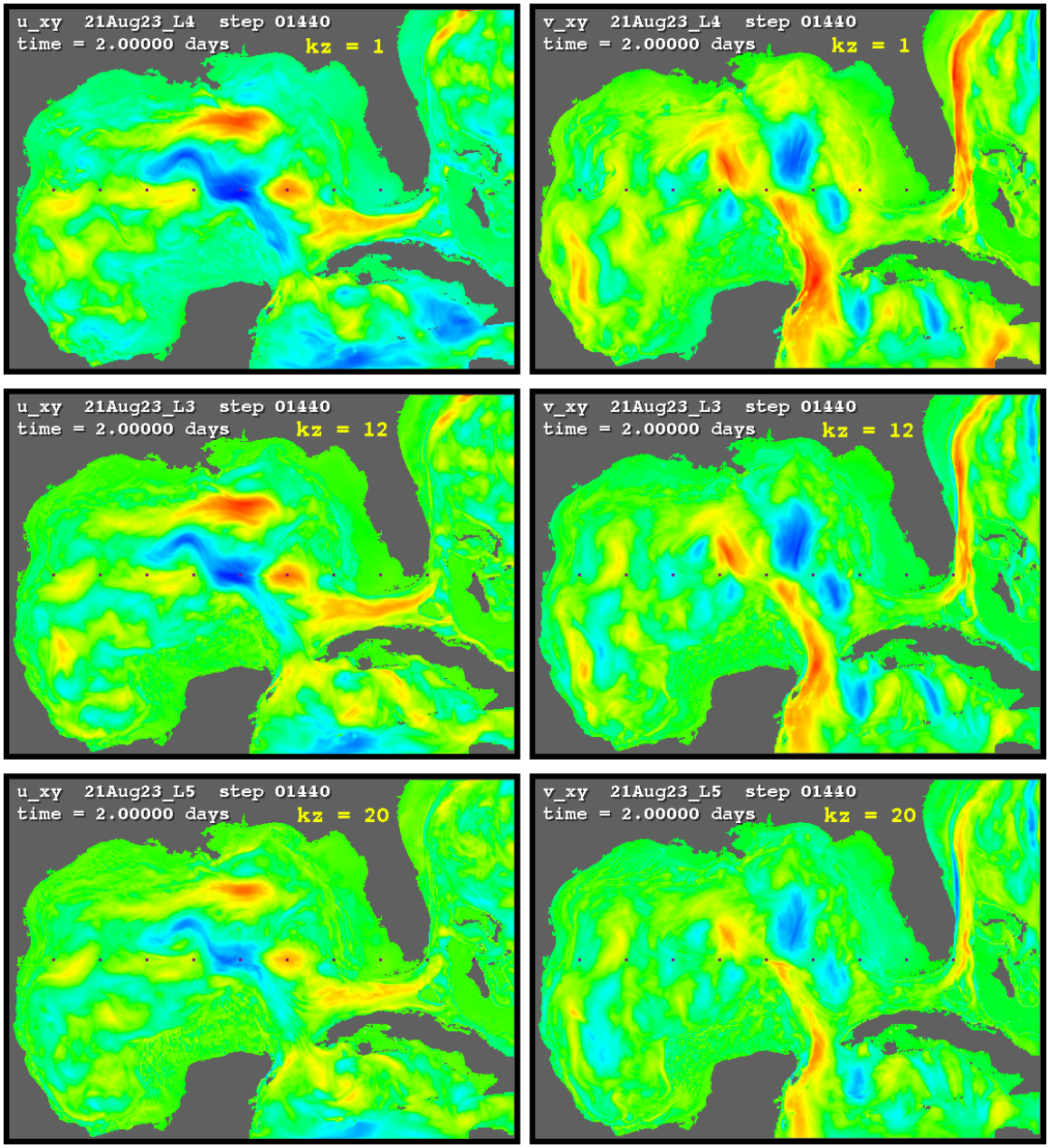

Fig 2. Horizontal flow velocities at 3 different z levels in the Gulf of Mexico after a 2-day HYCOM simulation (1440 timesteps). Left panels (top to bottom): East-west flow velocity u_xy (m/s) at three z levels k_z = 1, 12, and 20 of the 36 gridded levels. Right panels (top to bottom): North-south flow velocity v_xy (m/s) at the same three levels. The color map is a spectrum from dark blue, -1.45 m/s, to bright red, +1.45 m/s. File Fig 2. PT_step1440 3Z u_xy&v_xy.png

Figures 3 and 4 show the salinity (saln) and the temperature (temp in ^{0}C) at the three grid levels shown in Figs. 1 and 2 above. For each of the primary variables plotted in the four figures, the maximum and minimum values for the color maps were chosen to span the range of values found on level k_z = 12 throughout the run. These limits are held constant to simplify visually comparing the solutions at different times and places. However, the temperature at level k_z = 20, which is deeper than level 12, where the maximum and minimum values were determined, drops below the selected minimum plot value of 15^{0}C and a few locations to about 10.5^{0}C. The temperatures two days later on the right in Figs. 3 and 4, are exclusively between the preset limits of 15.0^{0}C and 33.0^{0}C on levels k_z = 1 and k_z = 12 over the entire grid.

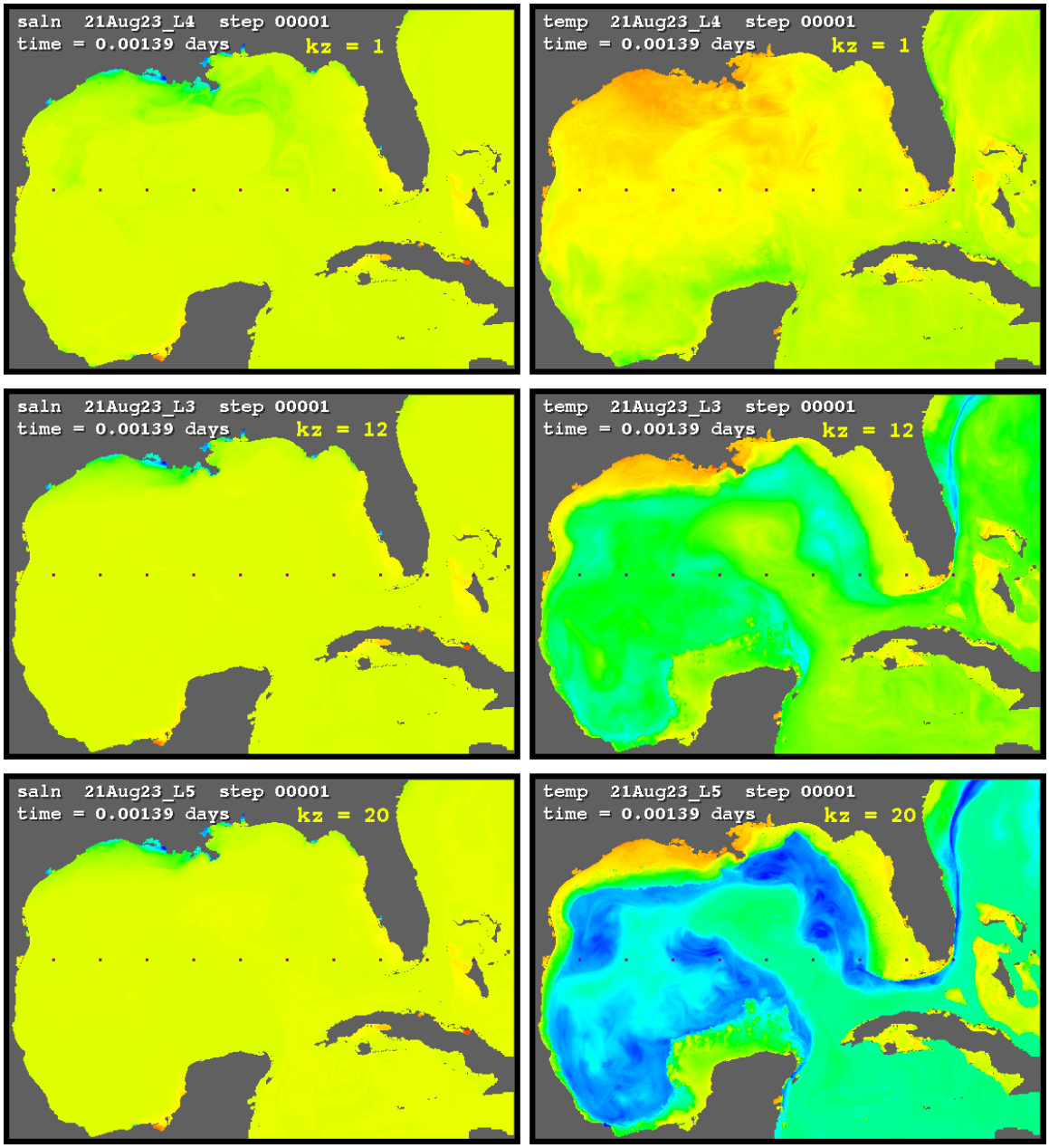

Fig 3. Salinity and temperature profiles at 3 different z levels in the Gulf of Mexico before the 2-day HYCOM simulation (at timestep 1). Left panels (top to bottom): Salinity (saln) (units?) at z levels 1, 12, and 20 of the 36 gridded levels. Right panels (top to bottom): Water temperature (^{0}C) at the same three levels. The color map for salinity is a spectrum from dark blue, 0.0 units, to bright red, 60 units. The color map for temperature ranges from dark blue, 15 ^{0}C, to bright red (35 ^{0}C). File Fig 3. PT_step1_3Z_saln&temp.png

Notice that the higher water temperatures in the righthand panels, indicted as yellow, orange and red, occur in the shallower water near shore. At the surface, the warmer temperatures spread over most of the Gulf but the temperatures drop quickly with depth in both the initial conditions in Fig. 3 above and the results after two days in Fig. 4 below. The minimum temperature on k_z = 20, in the lower right panels of Figs. 3 and 4 for the initial and final times respectively, are about 10.5^{0}C, indicated by the saturated dark blue patches well away from shore.

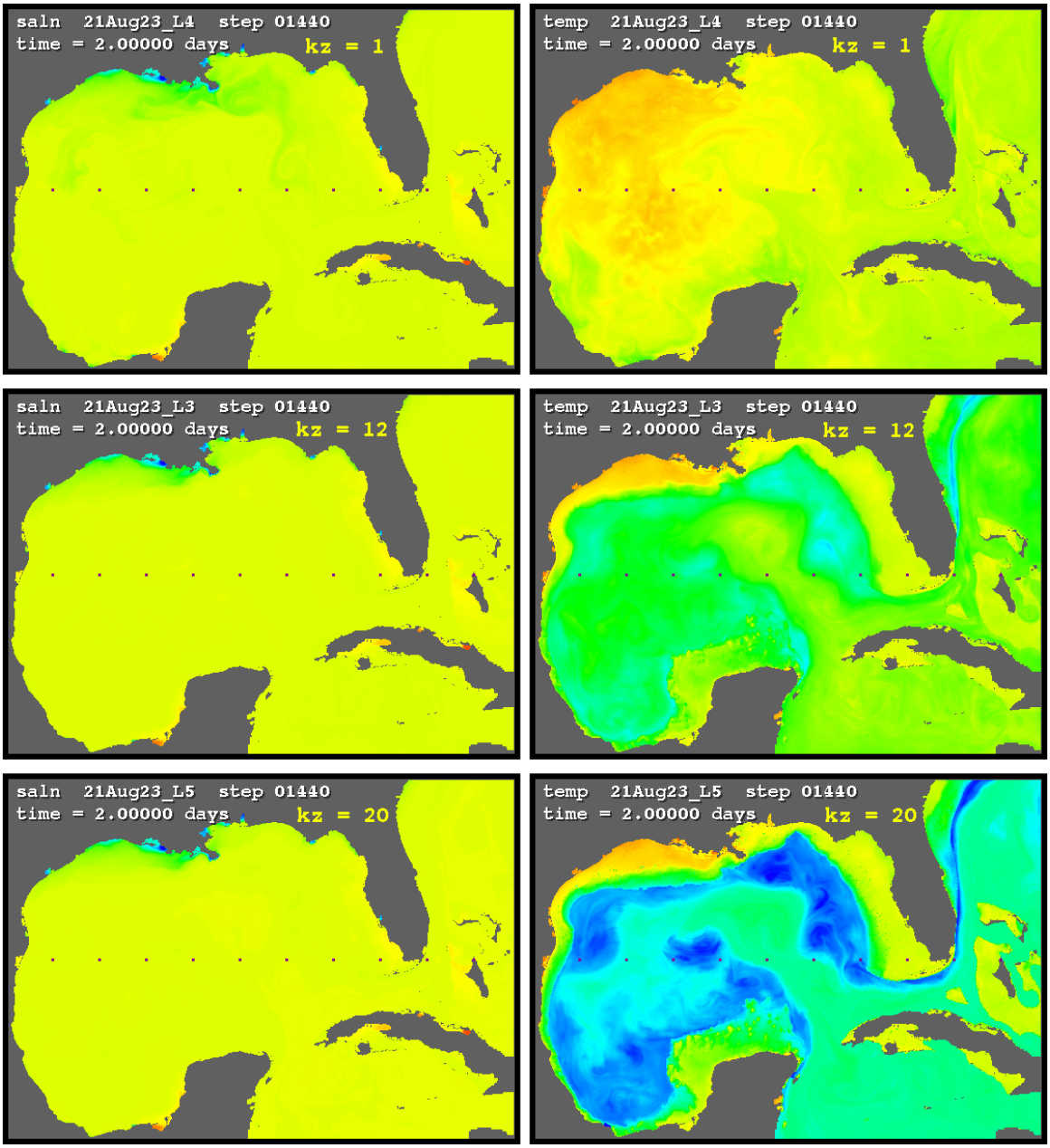

Fig 4. Salinity and temperature profiles at 3 different z levels in the Gulf of Mexico after a 2-day HYCOM simulation (1440 timesteps). Left panels (top to bottom): Salinity (saln) (units?) on z levels 1, 12, and 20 of the 36 gridded levels. Right panels (top to bottom): Water temperature (^{0}C) at the same three levels. The color map for salinity is a spectrum from dark blue, 0.0 units, to bright red, 60 units. The color map for temperature ranges from blue, 15^{0}C, to bright red (35^{0}C) with small dark blue regions at level k_z = 20 at about 10.5^{0}C. File Fig 4. PT_step1440_3Z_saln&temp.png

The fifth primary variable included in the current list of possible Diablo plots is the potential density. The potential density appears in HYCOM, named 'th3d,' and the concept also is used in atmospheric science. Potential density of the fluid is conserved as the pressure experienced by the parcel changes (provided no mixing with other parcels or net heat flux occurs). It is shown here in Fig. 5 to provide one illustration of 'th3d' variable. Figure 5 below shows an earth-map plot of the potential density 'th3d' deep in the Gulf at z level 20, after the 2-day (1440 timestep) run of HYCOM used to generate the figures above.

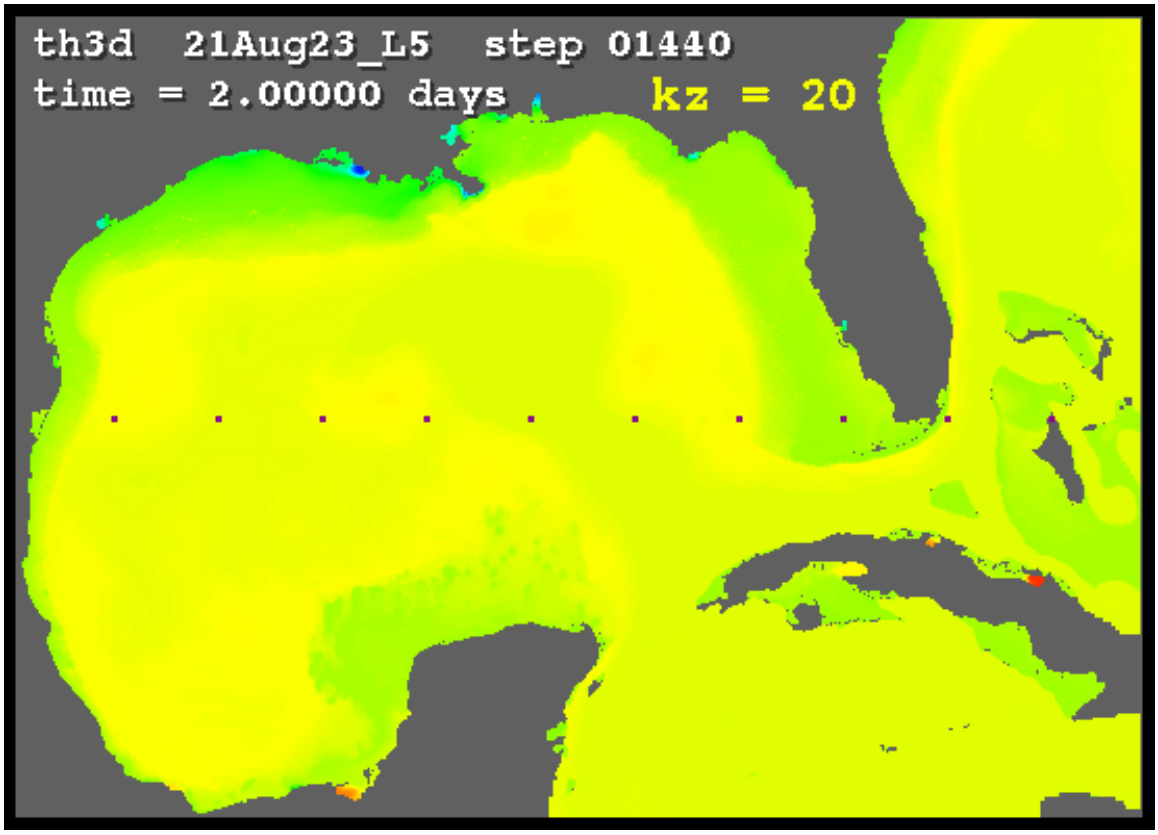

Fig 5. Potential density (th3d) at z level 20 in the Gulf of Mexico after a 2-day HYCOM run (1440 timesteps). The color map for th3d is a spectrum from dark blue, -31.0 units, to bright red, 11.0 units. File Fig 5. L5_th3d_20_001440.png

Diablo plots generally require a small fraction of the overall run time. The two runs 21Aug23_L2 and 21Aug23_L3, otherwise identical calculations, produced 490 Diablo plots and 90 plots respectively using plot frequencies of 30 and 180 steps. The additional 400 plots required an additional 6.6 seconds to compute, about .0165 seconds per plot. For run 21Aug23_L3, the 90 plots required less than .064% of the total run time. These plots are asynchronously transferred over VPN to the remote desktop, here a 2019 MacBook Pro, and thus the transfer and display does not slow the computation.

3. Configuration Timings for Gulf of Mexico Simulations on Luigi:

Table 1 following summarizes a large number of two-day HYCOM runs with different MPI - OpenMP hybrid parallel implementations. These are all identical 2-day Gulf of Mexico simulations. The 11Aug23 simulations at the top of the table, in black, were performed with a single rank running on a single Luigi node. With 128 cores available on the node, different numbers of OpenMP threads were used, resulting in very different total run times. We stopped at 16 cores since the core efficiency was dropping rapidly with more than 4 cores employed by OpenMP. The efficiency of the 1-thread case is defined as core efficiency 1.00, used as a normalization of all the other core efficiency estimates. The OpenMP core efficiency drops below 25% somewhere between 8 and 16 threads. This decline suggests opportunities for improvement.

Table1. Configuration Timings for Gulf of Mexico Simulations on Luigi								
run_ID	nodes	tasks/n	ranks	thrds/r	i & nplot	run times	core eff.	other data
11Aug23_L1	1	1	1	1	180	12.9s 7806s	1.000	scalar (1 core)
11Aug23_L2	1	1	1	2	180	14.0s 4556s	0.858	2 OMP thrds
11Aug23_L3	1	1	1	4	180	11.8s 3210s	0.609	4 OMP thrds
11Aug23_L4	1	1	1	8	180	12.0s 2421s	0.404	8 OMP thrds
11Aug23_L5	1	1	1	16	180	12.1s 2417s	0.203	16 OPM thrds
13Aug23_L4	1	4	4	16	180	9.3s 745.7s	.0826	node 0 kv=12
13Aug23_L5	1	4	4	8	180	9.5s 737.3s	.0837	node 0 kv=12
23Aug23_L1	1	4	4	4	180	11.6s 890.8s	.0692	node 25 kv=20
23Aug23_L2	1	4	4	2	180	9.4s 1313.0s	.0467	node 25 kv=12
23Aug23_L3	1	4	4	1	180	9.4s 2229.5s	.0274	node 25 kv=12
13Aug23_L6	1	8	8	8	180	9.3s 425.7s	0.146	
13Aug23_L7	1	8	8	16	180	9.3s 466.0s	0.133	
13Aug23_L8	1	8	8	12	180	9.3s 460.9s	0.135	
15Aug23_L9	2	8	16	16	180	9.1s 268.4	0.117	
15Aug23_LA	2	8	16	8	180	9.0s 256.8s	0.123	
15Aug23_L8	2	8	16	12	180	9.1s 258.40s	0.122	
15Aug23_LC	1	16	16	8	180	9.2s 294.6s	0.213	
15Aug23_LD	1	16	16	6	180	9.1s 324.0s	0.193	
15Aug23_L6	1	24	24	5	180	9.2s 286.8s	0.219	
15Aug23_L7	1	24	24	4	180	9.2s 288.4s	0.218	
15Aug23_L8	2	12	24	10	180	9.2s 255.6s	.0124	
15Aug23_L1	2	16	32	8	180	9.0s 189.8s	0.168	
15Aug23_L2	2	16	32	6	180	9.0s 202.8s	0.157	
15Aug23_L3	2	16	32	4	180	9.0s 204.4	0.157	
15Aug23_L4	1	32	32	4	180	9.4s 251.4s	0.252	
15Aug23_L5	1	32	32	3	180	9.2s 280.8s	0.218	
21Aug23_L2	1	32	32	4	30	9.1s 248.9s	0.254	node 15 kv=12
21Aug23_L3	1	32	32	4	180	9.1s 242.3s	0.261	node 15 kv=12
21Aug23_L4	1	32	32	4	180	9.1s 250.3s		node 15 kv = 1
21Aug23_L5	1	32	32	4	180	9.0s 248.6s		node 15 kv=20
15Aug23_LE	2	24	47	5	180	9.0 188.6s	0.169	
15Aug23_LF	1	48	47	2	180	9.1s 298.4s	0.210	
16Aug23_L2	4	16	63	8	30	18.3s 140.9s	0.124	
16Aug23_L1	4	16	63	8	180	20.1s 137.6s	0.130	
16Aug23_L3	3	21	63	6	180	19.2s 152.7s	0.152	
16Aug23_L4	2	32	63	4	180	15.8s 160.7s	0.210	
16Aug23_L6	2	32	63	4	30	17.7s 179.7s	0.188	
16Aug23_L5	2	32	63	2	180	19.2s 195.2s	0.173	
16Aug23_L7	1	64	63	2	180	s 258.4 s	0.268	
16Aug23_LA	4	32	127	4	180	9.9s 143.9s	0.056	
21Aug23_L1	3	43	127	3	180	12.4s 220.1s	0.098	
16Aug23_L9	2	64	127	2	180	11.3s 233.6s	0.137	
16Aug23_L8	1	128	127	1	180	28.0s 453.9s	0.143	

Following the 1-rank simulations at the top in black, are sets of runs, a different color for each different patch file used, giving a different number of ranks in the overall MPI partition. Each color depicts runs with a different number of ranks: 4 ranks in red, 8 ranks in red-orange, 16 ranks

in orange, 24 ranks in gold, 32 ranks in yellow-green (1 and 2 nodes), and 32 ranks in green (run on node 15 rather than node 0 with different z grid depth levels), 47 ranks in blue-green, 63 ranks in blue, and 127 ranks in purple/lavender at the bottom of the table. In each group of runs with the same number of ranks, different numbers of nodes and/or OpenMP threads were used. Increasing the number of cores reserved (more nodes) reduces the run's overall core efficiency.

The optimal core efficiency for each group is 'highlighted' in grey. Each optimum occurs when only one node is used. This means that more ensembles are possible with a given amount of computer resources (nodes) by running each ensemble member on its own node and not trying to run them any faster at reduced parallel efficiency. With 32 ranks, for example, two ensemble members can be computed in 251.4 seconds using two nodes working on separate ensemble members. Computing the same two ensemble members using 2 nodes together, first on one member, then on the second, would take 379.6 seconds.

For configurations with ranks from 16 to 63, the maximum core efficiency exceeds 20%. For 32-rank and 63-rank configurations the maximum efficiency exceeds 25%, again on a single node. Runs with 127 ranks (e.g., 32 tasks on each of 4 nodes) allow at most 4 cores for OpenMP threads, per task. The run, 16Aug23_LA in the last set of tabulated simulations above, with 4 threads for OpenMP, used barely 5% the 512 available cores for a core efficiency of 0.056. Increasing the number of tasks per node while decreasing the number of OpenMP threads can be seen to increase the core efficiency. Clearly further work is required on the OpenMP implementations.

4. More Timings for Gulf of Mexico Simulations on Luigi:

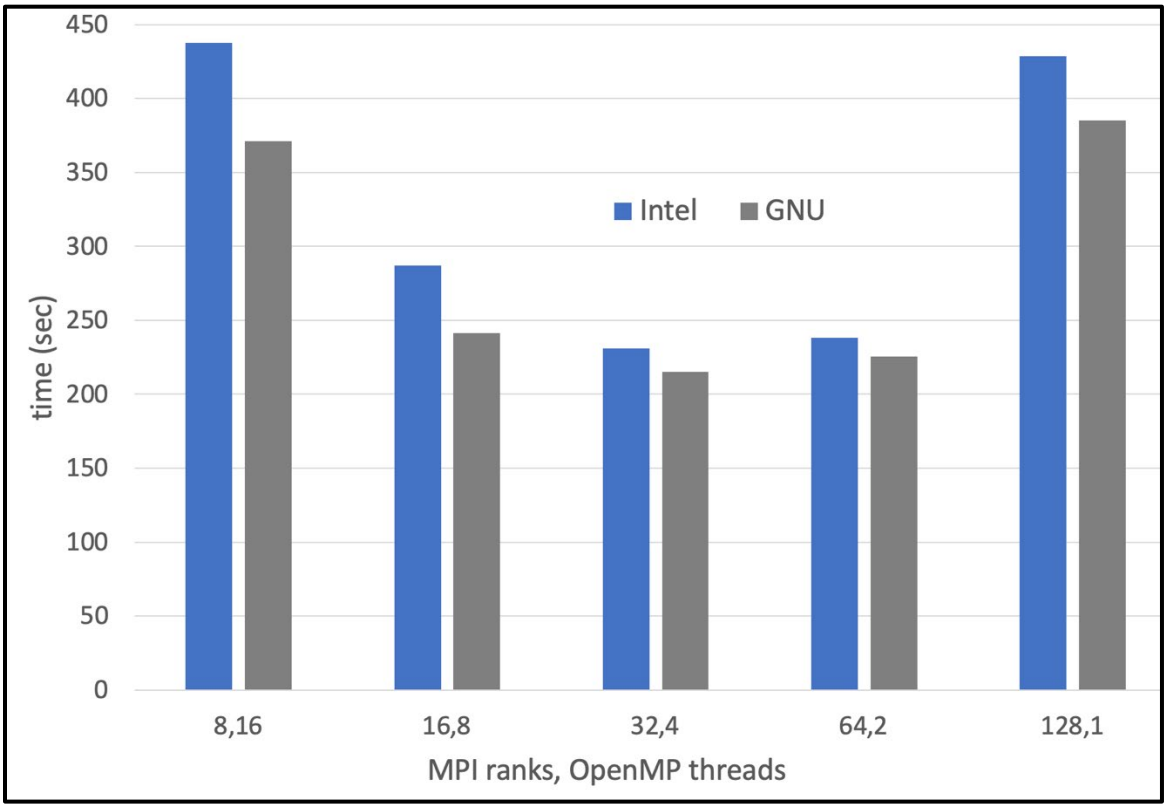

Fig. 6 Timings for the 2-day GOM case on one node of luigi with two different compilers. The number of ranks and threads multiply to 128, thus using all cores on the node. File Fig 6. Intel-GNU Runs.png

Figure 6, modified from a paper at the DoD HPC User Group Meeting (Obenschain, et al., 2023), shows results from another timing study of HYCOM using the Luigi system at NRL. These 2-day Gulf of Mexico runs, using two different compilers and only one node of Luigi, are summarized in Fig. 6. The possible combinations of MPI ranks and OpenMP threads are assigned to the runs in this study so an entire node of Luigi is used in each run. The GNU compiler provides the best timings while the Intel compiler results are very close to those of GNU. This figure shows that increasing the number of ranks, beyond a point reduces the overall parallel efficiency, presumably because larger fractions of the total rank data must be moved every timestep as the number of ranks increases. The optimum here is 32 MPI ranks with 4 OpenMP threads. If the OpenMP can be made more efficient, the overall performance will improve further.

Table 2 below summarizes a number of the two-day HYCOM runs with different MPI with OpenMP parallel implementations. These are all identical 2-day simulations. The 11Aug23 simulations at the top of the Table 1, in black, were performed with a single rank running on a single Luigi node. With 128 cores available on the node, different numbers of OpenMP threads were used, resulting in very different total run times. We stopped at 16 cores since the core efficiency was dropping rapidly with more than 4 dedicated for OpenMP optimization. The efficiency of the 1-thread case is defined as core efficiency 1.00 to normalize all other core efficiency estimates. The OpenMP core efficiency drops below 25% between 8 and 16 threads.

Comparing the Table 1 results above using the Intel compiler with the Intel-compiler results in Table 2 below shows very close agreement between the resulting run times and core efficiencies. This is to be expected because the more recent Table 1 results were using essentially the same system and compiler flags as the earlier results in Table 2. The table 1 results also include the small additional time required to preparing the Diablo plots every 180 timesteps. The 1-rank 1-thread normalization time used for Table 1 core-efficiency entries was also used in computing the core efficiencies for Table 2 because the corresponding 1-rank 1-thread times were not available for the GNU compiled runs.

Table 2. More Timings for Gulf of Mexico Simulations on Luigi

run_ID	nodes	tasks/n	ranks	thrds/r	i & nplot	run times	core eff.	other data
11Aug23_L1	1	1	1	1	180	12.9s 7806s	1.000	1 core (scalar)
13Aug23_L7	1	8	8	16	180	9.3s 466.0s	0.133	128 cores
15Aug23_LC	1	16	16	8	180	9.2s 294.6s	0.213	128 cores
21Aug23_L3	1	32	32	4	180	9.1s 242.3s	0.261	128 cores
16Aug23_L7	1	64	63	2	180	9.0s 258.4 s	0.268	128 cores
16Aug23_L8	1	128	127	1	180	9.6s 453.9s	0.134	128 cores
Intel cmplr	1	8	8	16	0 plots	9.8s 437s	0.142	128 cores
Intel cmplr	1	16	16	8	0 plots	9.2s 287s	0.219	128 cores
Intel cmplr	1	32	32	4	0 plots	9.1s 231s	0.274	128 cores
Intel cmplr	1	64	63	2	0 plots	9.1s 238s	0.278	128 cores
Intel cmplr	1	128	127	1	0 plots	9.6s 428s	0.145	128 cores
GNU cmplr	1	8	8	16	0 plots	9.8s 371s	0.168	128 cores
GNU cmplr	1	16	16	8	0 plots	9.2s 241s	0.262	128 cores
GNU cmplr	1	32	32	4	0 plots	9.1s 215s	0.296	128 cores
GNU cmplr	1	64	63	2	0 plots	9.1s 225s	0.282	128 cores
GNU cmplr	1	128	127	1	0 plots	9.6s 385s	0.163	128 cores

Notice that the 4 runs with 64 tasks actually used only 63 active ranks, meaning that 2 cores went unused. Similarly, the 4 runs with 128 tasks actually used only 127 ranks so that 1 core went

unused. In these two configurations one of the MPI tiles, apportioned by the 'patch' file, was totally over land and thus the corresponding rank had no computation.

Additional Gulf of Mexico Timings on Nautilus:

The Gulf of Mexico (GOM) case was also benchmarked on the Navy DSRC's recent addition, the Nautilus system. Nautilus is a Penguin HPCTrue system with 1304 standard nodes. Each node consists of an AMD EPYC Milan with 128 cores per node. The timings for 2-day GOM simulations using Intel compiler 2022.0.2 and penguin/openmpi/4.1.4/intel can be seen in Fig. 7. These timings are compared with those on Luigi system at NRL DC, reported in Fig. 6. Luigi is a 32-node system with AMD EPYC Rome processors. Each node has 128 processors and Intel compiler 19.1 and Intelmpi/19.1 are used on Luigi to run the same benchmark cases. We see that the timings on Nautilus are slightly faster than those on Luigi.

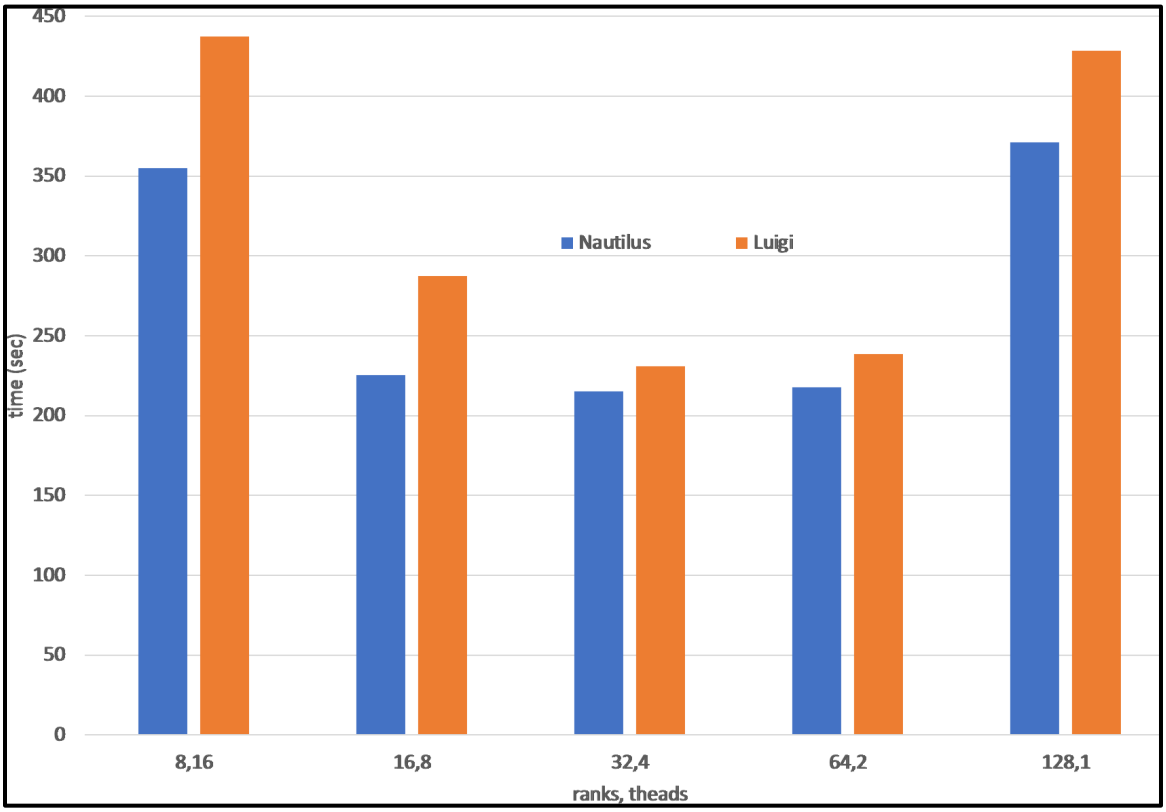

Fig. 7 Timings for the 2-day GOM case on one node of luigi (orange) and one node of nautilus (blue). The number of ranks and threads all multiply out to 128, thus using all cores on the node. File Fig 7. Nautilus Timings 27Nov2023.png

Data are not currently available to compute core efficiencies for these runs but the results are very similar to those of Fig. 6 and Tables 1 and 2. There seems to be an optimum performance around 32 ranks and four threads just as before. The top core efficiencies, where we have them, lie between 0.25 and 0.30. This means that at most a factor of three or four improvement would be possible if all the inefficiencies between MPI and OpenMP could be eliminated. We can aim for a factor of two.

5. Discussion and Conclusions:

The Gulf of Mexico test case being used is rather small and the computations of an optimal parallel-processing configuration and parameters will reflect this. A much larger or higher-resolution run would surely be optimized somewhat differently. In operational use an ensemble of simulations with different parameters is used to frame a prediction because input parameters and the time history of forcing functions are uncertain. The ocean ensemble spread stems from (i) accumulated differences in wind-forcing history and (ii) constraints of sea surface temperature by data assimilation. Differences in the specifics and parameterizations of the various physical sub-models provide a third source of end use uncertainty even when exactly the same models are used in each ensemble member. The intention of the ensemble is to reflect the actual uncertainty in initial conditions, which are largely unknown in terms of mesoscale circulation (Idzanovic, et al.,2023).

How many ensemble members are used for a particular operational forecast and how long is that forecast? How many more ensemble members would make a significant difference? What are the ranges of these numbers and what are the circumstances where the number needed can vary significantly? Can additional ensemble members for an existing forecast be provided later in response to changing circumstances? Maximum total running time would become a limitation in preparing operational ensembles and the number of ensemble members may have to be adjusted to complete the computations in an acceptable period.

More discussion of these issues is certainly warranted as well as considering trade-offs involving coarser spatial resolution for lower priority ensemble members to greatly increase the number of members to span wider parameter regimes. Here, we have considered the situation simplistically and await additional analysis when more operational information and options become available. Still, the message seems clear. Core efficiencies are higher when a single node is used, if the node is large enough for the runs, instead of two or more. This means that more ensemble members can be computed with a given total resource by using the nodes individually rather than working together to speed up the computation of each ensemble member.

The total computational resource available to prepare an operational forecast will be considered as a fixed limitation. If $\mathbf{N_E}$ is the number of equivalent ensemble members desired for the forecast, the computational resource has to be divided up to provide them. Many modern HPC clusters are hierarchical, having a number of relatively independent processing nodes that in themselves have a number of distinct processing constructs. For the simple discussion here, we will call the individual processing constructs ‘cores,’ and the independent processing nodes just ‘nodes.’ There are a significant number of HPC architectures today most of which have variants of this structure. In some the hierarchy can be deeper, or it may be more appropriate to call the cores ‘GPUs.’

Assume that $\boldsymbol{L}$ nodes, each having a number $\boldsymbol{K_C}$ of cores, can be assigned to each separate ensemble member computation and that they can be processed simultaneously. Such a situation is often called ‘embarrassingly parallel.’ Further, assume that computing each ensemble member is organized as a hybrid of MPI (Message Passing Interface) and OpenMP (Open Multi-Processing

or OMP) parallelization approaches. MPI divides an entire rectangular grid of N_i by N_j cells into m by n partitions or tiles, each of size m_x by n_y cells.

$$Total\ number\ of\ cells \equiv N_C = N_i N_j = m m_x n n_y$$

For a given resolution physical scenario, N_i and N_j, as well as N_C are fixed though m and n can be varied. Thus the questions become: What is the m by n partition that allows the most efficient computation of the ensemble members? What is the parallel processing configuration, including the MPI partition, that minimizes the computational time per 'core' reserved to compute a single ensemble member?

Above we noted that at most a factor of three or four speed up is possible if all the inefficiencies between MPI and OpenMP could be eliminated. OpenMP seems to be a good place to start. Several strategies have been proposed to do this. MPI also plays into this and simply longer rows and columns can help both MPI and OpenMP. When the number of MPI partitions is made larger for more parallelism, the size of each partition is smaller and boundary condition information that must be transferred by MPI to the neighboring partitions before computation become large enough that data transfer can dominate computation. One would rather begin to depend on OpenMP within large partitions. The time lost in data transfer would be reduced and the longer loop lengths within the partitions would tend to increase the efficiency of vectorization and OMP parallelization. The shape of the MPI partitions may also be worth optimizing. It can reduce the data transfer and leave longer loops within each partition for OMP to parallelize. Finally, it now appears that there are avenues to improve the performance of OpenMP. It seems that scaling could extend well beyond three or four threads.

Acknowledgements:

We wish to acknowledge the advice, help, and suggestions of colleagues who have worked with us in developing and implementing parallel fluid dynamics software on numerous generations of HPC hardware architectures at NRL. John Gardner, Fernando Grinstein, David Fyfe, Raafat Guirguis, Elaine Oran, Alexandra Landsberg, Chiping Li, Rainald Lohner and especially Rob Scott, Theodore Young Jr. We also wish to acknowledge the ONR/NRL base program for its steadfast support in performing the earlier research which has led to the capabilities and developments discussed here. This acknowledgement would be incomplete without highlighting the long-time NRL stalwart Steve Zalesak, whose multi-dimensional Flux-Corrected Transport algorithm is now enshrined in HYCOM for simulating the components of horizontal fluid dynamic transport.

HYCOM References:

Large, W.G., McWilliams, J.C. and Doney, S.C., (1994), "Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization," ***Rev. Geophys.***, 32: 363-403, 1994.

Bleck, R., (2002). "An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates," ***Ocean Modelling***, 4: 55-88.

A.J. Wallcraft, A.J., Metzger, E.J. and Carroll, S.N., (2009), "Software Design Description for the HYbrid Coordinate Ocean Model (HYCOM) Version 2.2," ***Naval Research Laboratory Memorandum Report*** NRL/MR/7320--09-9166, 12 February 2009.

Zalesak, S., (1979). "Fully multi-dimensional flux-corrected transport algorithms for fluids," ***J. Comput. Phys***., 31: 335-362.

Idzanovic , M., Rikardsen, E.S.U. and Röhrs, J. **, "**Forecast uncertainty and ensemble spread in surface currents from a regional ocean model," ***Front. Mar. Sci***. 10:1177337, Brief Research Report, 21 August 2023, DOI: 10.3389/fmars.2023.1177337.

Obenschain, K., Khine, Y. Y., Boris, J., Rosenberg, R., Patnaik, G. and Rowley, C., "Performance Study of the HYbrid Coordinate Ocean Model (HYCOM) using the HPCToolkit on an ARM System at NRL-DC," DoD HPC User Group Meeting 2023, US Army DEVCOM Army Research Laboratory, Aberdeen Proving Ground, MD, September 27-29, 2023.

Appendix A. Diablo Plots of Primary Variable Distributions

Diablo also can produce plots of the distribution of primary variable values in each of the earth-map plots shown in the body of this report. This is done by breaking the range between the minimum and the maximum value allowed for each primary variable into 100 bins and then counting the number of water cells in the earth-map plot with primary variable values that fall within those bins. This capability was added to study the small differences between solutions found using different parallelization configurations. An example of the distribution of horizontal east-west surface velocities, u(x,y) for the surface layer $k_z = 1$ is shown in Fig. 8. These are semi-log distribution plots.

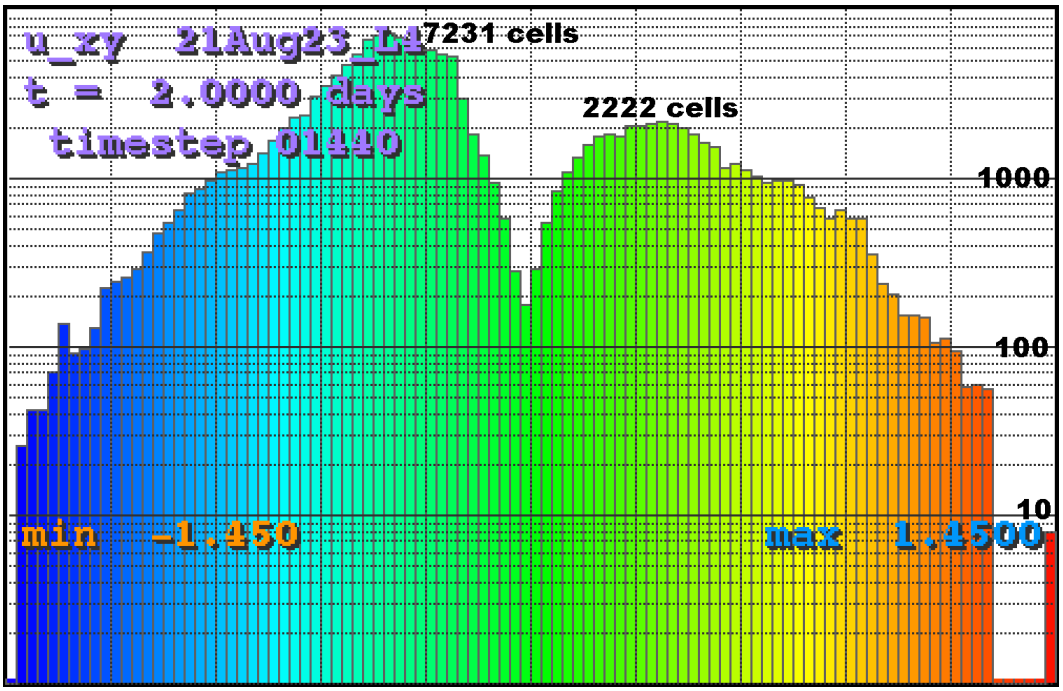

Fig 8. Distribution of east-west velocity u(x,y) values at Z level 1 in the Gulf of Mexico after a 2-day run (1440 timesteps) derived from the u_xy panel in the upper left corner of Fig. 2. The color map ranges from -1.45 m/s in bin 1 to +1.45 m/s in bin 100. File Fig 8. L4_ds_v_xy_1_001440.png

This example is double peaked as are most of the velocity distributions, both u(x,y) and v(x,y). The valley in the center appears because a square root rather than a linear scaling is used between the minimum and the maximum to determine the color to plot for a given value. This was done to emphasize the regions of high and low speed flow as seen in Figs. 1 and 3 with greater color variation. The square root scaling was also used for the salinity because almost all deviations from a near constant salinity occur along the shores, as seen in Figs. 3 and 4. The linear scaling is used for the temperature (temp) and the potential density (th3d).

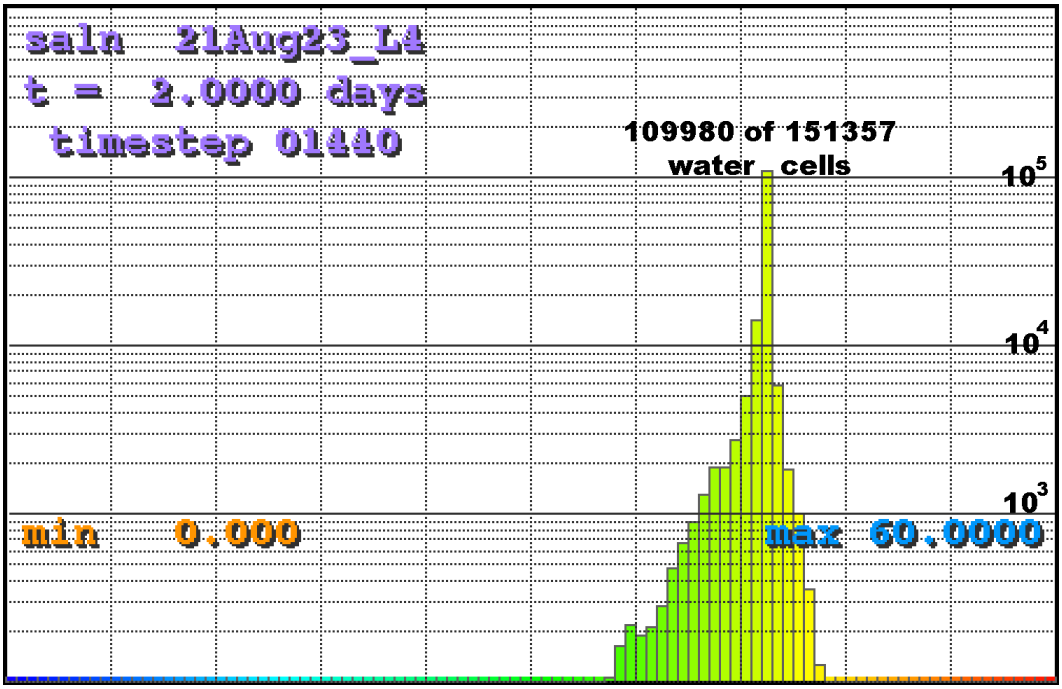

Fig 9. Salinity distribution at z level 1 in the Gulf of Mexico after a 2-day run (1440 timesteps). The color map for salinity is a spectrum from dark blue, 0.0 units, to bright red, 60 units. This highly peaked distribution comes from upper left panel of Fig. 4. File Fig 9. L4_ds_saln_1_001440.png

Diablo, as currently implemented in HYCOM, can produce not only the primary variable earth map plots shown in Figs. 1 through 4 in Section 2 above but also produces corresponding plots of a fifth primary variable, the potential density 'th3d,' when required. One such earth-map plot is illustrated in Fig. 10 below.

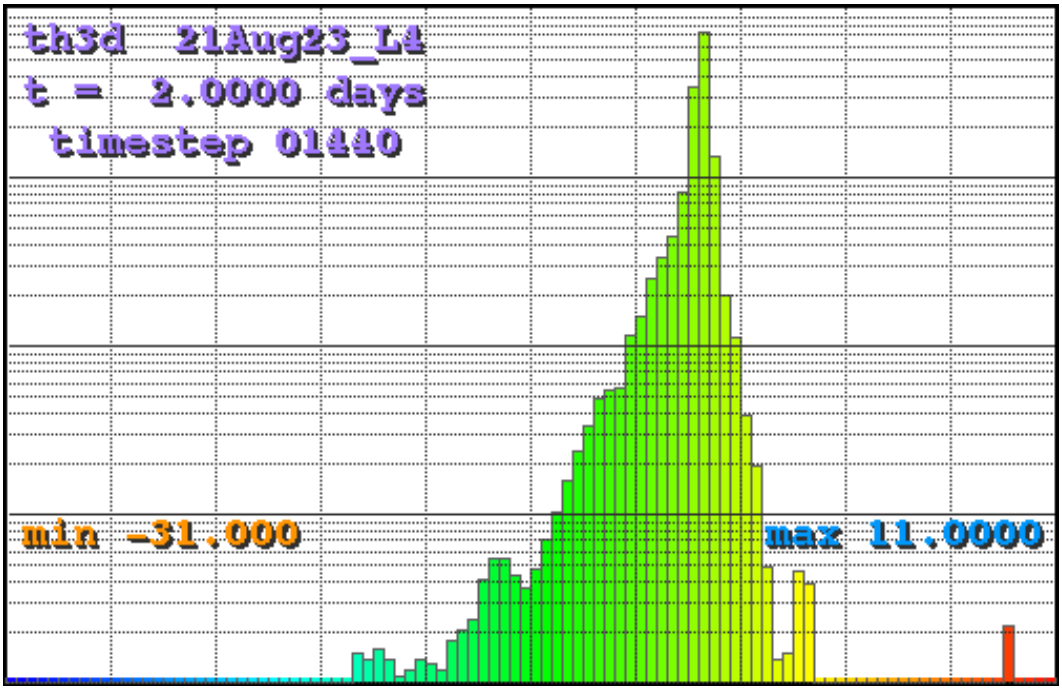

Fig 10. Potential density ('th3d') distribution at z level 1 in the Gulf of Mexico after a 2-day run (1440 timesteps). The color map for 'th3d' is a spectrum from dark blue, -31.0 units, to bright red, 11.0 units. File Fig 10. L4_ds_th3d_1_001440