ASC Student Supercomputer Challenge 2026

Preliminary Round Official Announcement

Dear ASC26 Teams:

Welcome to the 2026 ASC Student Supercomputer Challenge (ASC26)!

Entering its 13th edition, the ASC Student Supercomputer Challenge continues to grow as the world's largest supercomputing hackathon, gathering passionate students from around the globe to explore, innovate, and push the boundaries of high-performance computing and artificial intelligence. Since its launch on November 6, 2025, ASC26 has generated overwhelming enthusiasm, with hundreds of outstanding student teams registering to take on this year's challenge. As the competition now moves into the Preliminary Phase, your ASC26 journey truly begins, a journey of discovery, teamwork, and technical excellence.

Preliminary Round: Time to Design, Optimize, and Submit.

In the ASC26 Preliminary Round, teams are encouraged to give their very best as they tackle the assigned challenges and submit a comprehensive proposal. Your submission should clearly present your cluster design, source code optimization strategies, and output results, demonstrating both technical rigor and creative problem-solving. All proposals, submitted in English, will be carefully and rigorously reviewed by the ASC26 Evaluation Committee, ensuring a fair, transparent, and thorough assessment of every team's work.

Important Note: All submitted proposals will undergo a plagiarism check by the ASC26 Evaluation Committee. If substantial similarities are found between any two submissions, both teams will receive a score of zero. We therefore encourage every team to work independently, to think creatively, and to showcase your own technical abilities and original ideas. Academic integrity is a core value of ASC, and all teams are expected to uphold the highest standards by refraining from any form of plagiarism.

Submission Guidelines.

Important Deadline: All participating teams must submit the required materials by 24:00 on March 4, 2026 (UTC/GMT +8:00).

a) **Proposal Submission:** Each team is required to submit a proposal that includes a clear and properly structured table of contents. The proposal file must be named using the following format: [University/College Name]_[Contact Person Name] (e.g., ABC_University_John_Doe). The complete proposal must be compiled into one single PDF file and uploaded to the official submission platform at ASC26 website:

https://www.asc-events.net/StudentChallenge/ASC26/PreliminarySubmission.php

b) **Additional Materials**: All supplementary materials must be compressed into a single file and named using the following convention: [University/College Name]_[Contact Person Name]. The compressed file must contain no fewer than four folders, structured as specified in Appendix A. The compressed file must be uploaded to the designated FTP server. Access details, including the FTP address, will be provided to teams via email at a later date.

- Output files of HPL
- Output files of HPCG
- Required files of Embodied World Model
- Required files of AMSS-NCKU

c) **Submission Confirmation and Support:** All required materials must be completed and submitted by the specified deadline and in full compliance with the submission guidelines. Submissions that are incomplete, improperly formatted, or submitted after the deadline will not be evaluated and will receive no score. A confirmation email will be sent shortly after all required materials have been successfully received. For further inquiries or assistance, please contact:

- Technical Support: techsupport@asc-events.org
- General Information: info@asc-events.org
- Press: media@asc-events.org

The ASC26 Organizing Committee extends its best wishes to all participating teams as they embark on ASC26 journey.

Appendix A:

Proposal Requirements.

I. Introduction to the university's activities in supercomputing (5 points)

1. Supercomputing-related hardware and software platforms.
2. Supercomputing-related courses, training programs, and interest groups.
3. Supercomputing-related research and applications.
4. A brief description of up to two key achievements in supercomputing research
5. This section should be no more than 3 pages in length.

II. Team introduction (5 points)

1. A brief description of the team setup.
2. Introduction and photo of each member, including group photos of the team.

3. Team motto or catchphrase.
4. This section should be no more than 5 pages in length.

III. Technical proposal requirements (90 points)

1. Design of AI&HPC system (15 points)

Participating teams are required to submit a theoretical design of an AI and HPC cluster, rather than construct a physical system. The proposed cluster design must comply with the following requirements:

a) **Computing Performance and Power Constraints:** The system must be designed to achieve optimal computing performance. The cluster must consist of no fewer than three compute nodes, with a maximum power consumption of 2,000 W per node and a total power limit of 5,000 W for all cluster components combined.

b) **System Configuration and Architecture Analysis**: Teams must clearly specify the system's hardware and software configurations, as well as the interconnection architecture. The proposal must include a description of power consumption, an evaluation of system performance, and an analysis of the advantages and limitations of the proposed architecture.

c) **Architectural Diagram and Power Consumption Documentation**: This section must include an architectural diagram of the system designed by the team, with the power consumption of each major component clearly indicated within the diagram. The total length of this section must not exceed 15 pages.

Item	Name	Configuration
Server	Dual Processor Server	CPU：AMD 9755-EPYC2.7_128C_512M * 2 Memory: 32GB * 24, DDR5, 6400 MT/s Hard disk: 960GB SSD SATA * 1
HCA	NDR400	NVIDIA MCX75310AAS-NEAT
Switch	Ethernet switch	10/100/1000 MB/s，24 ports Ethernet switch
	NDR-IB switch	NVIDIA Quantum (TM)-2 NDR InfiniBand Switch, 64-ports NDR, 32 OSFP ports, unmanaged, P2C airflow (forward)
Cable	Gigabit CAT6 cables	CAT6 copper cable, blue, 3 m
	InfiniBand cable	InfiniBand NDR copper cable, OSFP port, compatible with the InfiniBand switch in use.

Notes:

(1) The hardware components listed in the table above are provided for reference purposes only.

(2) The reference configuration is based on a dual-processor server architecture that supports up to two GPUs.

(3) The hardware configuration used during the ASC26 Competition Finals may differ from the reference configuration listed in the table above.

(4) The sponsors of the ASC26 Finals will provide AMD W7900D GPUs for use by all participating teams. Additionally, in the preliminary stage, a remote testing platform equipped with AMD W7900D GPUs will be made available. The login information will be sent to all participants via email at a later time.

2. HPL and HPCG Benchmarks (15 points)

The proposal must include a detailed description of the software environment, including but not limited to the operating system, compiler, mathematical libraries, MPI implementation, and software versions. Teams must also describe their performance optimization and testing methodologies, performance measurement approaches, and problem identification and solution analysis. An in-depth analysis of the HPL and HPCG algorithms, including discussion of the relevant source code implementations, will be considered a strong advantage in the evaluation process.

Download the HPL software at: http://www.netlib.org/benchmark/hpl/

Download the HPCG software at: https://github.com/hpcg-benchmark/hpcg

Teams are strongly encouraged to perform verification and performance optimization of the HPL and HPCG benchmarks on x86 CPU and data center GPU platforms. If alternative hardware platforms are used, teams may submit the corresponding performance analysis and results, provided that these materials clearly demonstrate adequate performance.

The output results of HPL and HPCG must be saved in directories named “HPL” and “HPCG”, respectively, and uploaded to the designated FTP server as part of the submission materials.

3. Embodied World Model Optimization (30 points)

Embodied Intelligence represents a transformative frontier in artificial intelligence, integrating perception, reasoning, and physical action within unified systems. By enabling intelligent agents to interact directly with the physical world, embodied intelligence bridges the gap between digital computation and real-world agency. General-purpose humanoid robots occupy a central position in this field, as they must operate effectively in unstructured environments, manipulate a wide range of objects, and demonstrate a practical understanding of physical principles.

For a humanoid robot to act purposefully, effective operation depends on a tight coupling between decision-making and environmental simulation. The Vision–Language–Action (VLA) model functions as the agent’s cognitive core, translating linguistic instructions and perceptual inputs into concrete action proposals. Complementing this component is the Learned World Model, which serves as a neural simulator of the environment. By evaluating the candidate actions generated by the VLA, the World Model predicts future outcomes through the generation of physically realistic video sequences. This form of "mental rehearsal" enables the creation of a physically plausible digital twin of the real world, supporting policy evaluation, and safety validation without the risks associated with physical trial-and-error. However, in contrast to the discrete action outputs of the VLA, the World Model must synthesize high-dimensional, continuous video representations, making it the dominant computational bottleneck within the overall system.

To address this computational bottleneck and enable practical deployment, the Preliminary Round of this challenge focuses on accelerating UnifoLM-WMA-0, a state-of-the-art world model developed by Unitree Robotics for embodied intelligence tasks. Participants are required to optimize the model's inference performance under specified scenarios and command inputs, thereby enabling rapid "mental rehearsal" for the robot, while strictly maintaining a defined visual quality threshold.

The task will be evaluated using five robotic scenarios provided by the Committee. From each scenario, four representative samples are selected, resulting in a total of 20 evaluation samples. The inference time measured for each sample will serve as the primary quantitative performance metric. Peak Signal-to-Noise Ratio (PSNR) will be used to assess the visual fidelity between the videos generated by participants and the corresponding reference videos provided by the Committee. For each of the 20 evaluation samples, the generated video must achieve a PSNR of at least 25. If any sample within a given scenario fails to meet the PSNR $\geqslant$ 25 requirement, the scenario will receive zero points.

Workflow Tutorial

(1) Install UnifoLM-WMA-0 project and download the model checkpoints

UnifoLM-WMA-0 can be accessed through the GitHub repository https://github.com/unitreerobotics/unifolm-world-model-action. And the model checkpoints of UnifoLM-WMA-0 can be obtained from: https://huggingface.co/unitreerobotics/UnifoLM-WMA-0-Dual

(2) Download the input files

Download the initial conditions and reference videos for 5 scenarios with 20 samples as input files from the ASC repository https://github.com/ASC-Competition and place them in UnifoLM-WMA-0 repository. The following is an example of the input file structure for a single scenario:

```
UnifoLM-WMA-0/unitree_g1_pack_camera
    ├── case1
    |   ├── run_world_model_interaction.sh
    |   ├── unitree_g1_pack_camera_case1.mp4
    |   └── world_model_interaction_prompts
    |       ├── images
    |       |   └── unitree_g1_pack_camera
    |       |       └── 0.png
```

```
|           ├── transitions
|           |    └── unitree_g1_pack_camera
|           |         ├── 0.h5
|           |         └── meta_data
|           |              └── stats.safetensors
|           └── unitree_g1_pack_camera.csv
├── case2
├── case3
└── case4
```

(3) Run inference

Specify the correct absolute path for data_dir in configs/inference/world_model_interaction.yaml according to the README of https://github.com/unitreerobotics/unifolm-world-model-action.

Once the above preparations are complete, you can execute the inference task using the following command:

```
cd unifolm-world-model-action
bash {scenario_name}/{case_id}/run_world_model_interaction.sh
# For example: bash unitree_g1_pack_camera/case1/run_world_model_interaction.sh
```

Upon execution, the script will produce output similar to the following:

```
  0%|
| 0/8 [00:00<?, ?it/s]>>> Step 0: generating actions ...
>>> Step 0: interacting with world model ...
>>>>>>>>>>>>>>>>>>>>>>>>>
 12%|███████████████████████████████▍
| 1/8 [00:46<05:22, 46.03s/it]>>> Step 1: generating actions ...
>>> Step 1: interacting with world model ...
>>>>>>>>>>>>>>>>>>>>>>>>>
 25%|█████████████████████████████████████
| 2/8 [01:33<04:39, 46.65s/it]>>> Step 2: generating actions
...
...
>>> Step 7: interacting with world model ...
>>>>>>>>>>>>>>>>>>>>>>>>>
100%|█████████████████████████████████████
8/8 [06:10<00:00, 46.29s/it]

real      8m15.226s
user      278m50.952s
```

```
sys        17m21.528s
```

In this challenge, the real time is considered as the inference time, which is 8m15.226s for 1 sample from unitree_g1_pack_camera dataset.

Upon completion of the inference process, an output directory will be generated. Below is an example of the directory structure obtained from the inference of one case from unitree_g1_pack_camera dataset:

```
UnifoLM-WMA-0/unitree_g1_pack_camera
    ├── case1
    │   ├── output
    │   │   ├── inference
    │   │   │   ├── 0_full_fs6.mp4
    │   │   │   └── sample_0
    │   │   │       ├── dm
    │   │   │       │   └── 6
    │   │   │       │       ├── itr-0.mp4
    │   │   │       │       ├── itr-1.mp4
    │   │   │       │       ├── itr-10.mp4
    │   │   │       │       ├── itr-2.mp4
    │   │   │       │       ├── itr-3.mp4
    │   │   │       │       ├── itr-4.mp4
    │   │   │       │       ├── itr-5.mp4
    │   │   │       │       ├── itr-6.mp4
    │   │   │       │       ├── itr-7.mp4
    │   │   │       │       ├── itr-8.mp4
    │   │   │       │       └── itr-9.mp4
    │   │   │       └── wm
    │   │   │           └── 6
    │   │   │               ├── itr-0.mp4
    │   │   │               ├── itr-1.mp4
    │   │   │               ├── itr-10.mp4
    │   │   │               ├── itr-2.mp4
    │   │   │               ├── itr-3.mp4
    │   │   │               ├── itr-4.mp4
    │   │   │               ├── itr-5.mp4
    │   │   │               ├── itr-6.mp4
    │   │   │               ├── itr-7.mp4
    │   │   │               ├── itr-8.mp4
    │   │   │               └── itr-9.mp4
    │   │   └── tensorboard
    │   │       └── events.out.tfevents.1767167731.e85c9c312cc5.16758.0
    │   └── output.log
    ├── case2
    ├── case3
    └── case4
```

(4) PSNR evaluation

Download the PSNR evaluation script and the reference videos from the ASC repository https://github.com/ASC-Competition. Then, execute the PSNR calculation script to obtain the output:

```
python3 psnr_score_for_challenge.py --gt_video path/to/gt/videos --pred_video path/to/pred/video --output_file path/to/output/json
# For example: python3 psnr_score_for_challenge.py --gt_video unitree_g1_pack_camera/case1/unitree_g1_pack_camera_case1.mp4 --pred_video unitree_g1_pack_camera/case1/output/inference/0_full_fs6.mp4 --output_file unitree_g1_pack_camera/case1/psnr_result.json
```

This script will generate a JSON file containing a dictionary of the PSNR score for one sample. An example of the JSON file content is (PSNR score shown in this example are for illustration purposes only and should not be used as a reference.):

```
{
    "gt_video": "unitree_g1_pack_camera/case1/unitree_g1_pack_camera_case1.mp4",
    "pred_video": "unitree_g1_pack_camera/case1/output/inference/0_full_fs6.mp4",
    "psnr": 28.32714035028894
}
```

Result Submission

Please submit all the requested files by compressing them in a file named Teamname_worldmodel.tar.gz. Below is the directory structure:

```
Teamname_worldmodel
    ├── code
    ├── proposal_file
    └── results
        ├── summary.json
        ├── unitree_g1_pack_camera
        |   ├── case1
        |   |   ├── 0_full_fs6.mp4
        |   |   └── output.log
        |   ├── case2
        |   ├── case3
        |   └── case4
        ├── unitree_z1_dual_arm_cleanup_pencils
        |   ├── case1
        |   ├── case2
        |   ├── case3
        |   └── case4
        ├── unitree_z1_dual_arm_stackbox
        |   ├── case1
        |   ├── case2
```

```
|     ├──── case3
|     └──── case4
├──── unitree_z1_dual_arm_stackbox_v2
|     ├──── case1
|     ├──── case2
|     ├──── case3
|     └──── case4
└──── unitree_z1_stackbox
      ├──── case1
      ├──── case2
      ├──── case3
      └──── case4
```

Notes:

(1) The complete source code of the project should be provided.

(2) 20 directories for 20 cases of 5 scenarios should be provided.
For each case, the case directory must contain the complete MP4 video file (such as 0_full_fs6.mp4) and the output.log file containing the screen output including real time during inference procedure.

(3) A summary.json file should be provided.
Please generate the JSON file strictly in accordance with the specified format. The time and PSNR values shown in the example are provided for illustrative purposes only, and they must not be used as a reference for evaluation.

```
[
        {"scenario": "unitree_g1_pack_camera",
        "case_id": 1,
        "time_before_optim": 606.984,
        "time_after_optim": 397.532,
        "psnr": 28.3627},
        {"scenario": "unitree_g1_pack_camera",
        "case_id": 2,
        "time_before_optim": 601.228,
        "time_after_optim": 408.252,
        "psnr": 26.1782},
        ......
        {"scenario": "unitree_z1_stackbox",
        "case_id": 4,
        "time_before_optim": 542.386,
        "time_after_optim": 338.097,
        "psnr": 29.2963}
          ]
```

The JSON file must contain a list of 20 dictionaries, each containing the result information for a case. Each dictionary includes the scenario name, case ID, inference time before and after optimization (in seconds), and the PSNR score.

4) A proposal file should be provided.
The proposal file may be submitted in DOC or PDF format and must include the evaluation results, a comprehensive description of the optimization methods, and a comparison of execution time before and after optimization for all evaluation samples.

Evaluation Criteria Overview

During the evaluation process, the ASC26 Committee will place primary emphasis on improvements in inference performance. In addition, proposals that present clear, well-justified optimization strategies together with a sound explanation of the underlying technical principles will serve as a key basis for scoring. The following points highlight important considerations for participating teams.

(1) The model's weight parameters are restricted to a minimum of 16-bit precision; sub-16-bit quantization of weights is not allowed.

(2) Do not modify the run_world_model_interaction.sh file in each case directory (except for CUDA_VISIBLE_DEVICES). Do not modify the configs/inference/world_model_interaction.yaml file (except for data_dir).

(3) Do not modify the provided input files.

(4) Please keep the directory structure and the resulting files unchanged.

(5) For each of the 20 samples, the generated video must achieve a PSNR ≥ 25.

(6) The submitted inference log files should contain details such as time consumption for each step and other essential elements during the inference process.

(7) Participants are required to provide comprehensive details of model inference process, the machine specification, the environment setup, the optimization methods used and the performance comparison, which will be considered as the primary basis for scoring by the ASC26 committee.

(8) Participants must submit all required files as specified above. Failure to provide any of the required materials will result in a score of zero for this component of the evaluation.

4. Numerical Relativity Code AMSS-NCKU Optimization (30 points)

General relativity is the fundamental theory describing gravity. Although it has been established for more than a century, our understanding of general relativity remains limited. One of the primary reasons is that its governing dynamical equation, the Einstein field equations, constitutes one of the most challenging systems of partial differential equations in physics. To analyze gravitational wave detection data, it is essential to obtain accurate solutions describing binary black hole mergers.

Gravitational wave detection was recognized with the 2017 Nobel Prize in Physics, following the landmark first observation in 2015. This breakthrough was made possible in large part by numerical solutions of the Einstein field equations, which provided the theoretical waveforms essential for signal identification and parameter estimation. Ali, a gravitational wave detector currently under construction and operation in China, is designed to extend observational capabilities into the 10^{-18} Hz frequency band, opening access to previously unexplored gravitational wave signals. With the availability of such observations, researchers aim to probe the earliest epochs of the universe, offering new insights into the physics of the Big Bang and the fundamental nature of spacetime, further establishing gravitational wave science as a frontier application for modern high-performance computing.

Software used to solve the Einstein field equations numerically is generally referred to as a numerical relativity code. Several numerical relativity codes have been developed worldwide, although only a limited number are publicly available. AMSS-NCKU is one such open and publicly accessible numerical relativity code. AMSS-NCKU employs finite-difference methods to solve the Einstein field equations. To address the multi-scale nature inherent in general relativity, the code utilizes adaptive mesh refinement (AMR) techniques. For parallel computing, AMSS-NCKU adopts a domain decomposition strategy combined with MPI-based parallelization. The computational workflow of AMSS-NCKU proceeds in two main stages. First, the code solves the elliptic components of the Einstein field equations to generate initial data. Then AMSS-NCKU solves the hyperbolic part of Einstein equation to get the spacetime metric for the following times. The solution of the hyperbolic evolution equations constitutes the dominant computational workload of AMSS-NCKU.

With respect to the hyperbolic evolution component of AMSS-NCKU, two computational blocks are of primary importance. The first consists of the Runge–Kutta time-integration steps used to evolve the Einstein field equations. The second involves data communication and synchronization across multiple mesh refinement levels and distributed MPI processes. In contrast to typical computational fluid dynamics (CFD) applications, the Einstein field equations involve several tens of unknown variables. Moreover, the spacetime metric fields are generally smooth and continuous, which further distinguishes numerical relativity from fluid dynamics problems.

The objective of this task is to minimize the total elapsed execution time, reported as "This Program Cost = ** Seconds", which will be printed to the screen upon completion of the simulation.

Workflow Tutorial

The numerical relativity simulations with AMSS-NCKU can be initiated via the python script, which is available in the github repository https://github.com/ASC-Competition. After cloning the repository, you will find the AMSS-NCKU source files (the numerical relativity simulation code) and the required Python files (the interface and script files for running simulations). Here is an example of the AMSS-NCKU file directory:

NR-amssncku/

|____amssncku

|___ AMSS_NCKU_ sources/ (The AMSS-NCKU C++ and Fortran source files)

|___ inputfile_examples/

|___ makefile_influde/ (Useful Examples to compile the AMSS-NCKU code)

|___AMSS_NCKU_Program.py (The Python script file that is used to start an AMSS NCKU simulation)

|___AMSS_NCKU_Input.py (The physical parameters are set in this script file)

|___......

To perform a binary black hole simulation, follow these steps:

1. **Install prerequisites**: Ensure that gcc, gfortran, nvcc, and python3 are installed. Modify the makefile.inc file in the AMSS_NCKU_sources folder to match your test platform, using the examples in makefile_include as a reference. Further details are available in the repository.
2. **Configure input parameters**: In the file AMSS_NCKU_Input.py, NO modifications are permitted to the methods or physical parameters, with the sole exception of the parallelization parameters.
3. **Build and run the simulation**:
   Execute the following command in a terminal:
   $ python3 AMSS_NCKU_Program.py
   The code will automatically generate all output files for the binary black hole simulation.

Example

Here we present a typical binary black hole merger simulation example. It corresponds to the first observed gravitational wave event --- GW150914 --- captured by LIGO and VIRGO collaborations in 2015. This gravitational wave event is generated by a binary black hole system with masses $M_1 = 36M_{\odot}$ and $M_2 = 29M_{\odot}$. When these two massive black holes spiraled towards each other and eventually merged, they produced an extremely strong gravitational wave signal. The GW150914 made a groundbreaking discovery, for it not only provides direct confirmation of Einstein's prediction of gravitational waves, but also opens up a new window for observing our universe via the gravitational wave astronomy.

In the simulation, AMSS-NCKU solves the elliptic part of Einstein equation using a “Two Puncture” method and obtains the initial data for spacetime variables at t=0. Then the hyperbolic part of Einstein equation is evaluated using the adaptive mesh refinement techniques and finite difference schemes to solve various variables on spatial grids (to reduce computational cost, a 4^{th}-order finite difference scheme is used rather than a higher-order finite difference scheme), which dominates the time evolution of the binary black hole system. The Runge-Kutta method is used in the simulation for time integration. When the AMSS-NCKU simulation for GW150914 event is successfully done, all output files in this simulation are stored in the output file directory, with the following structure:

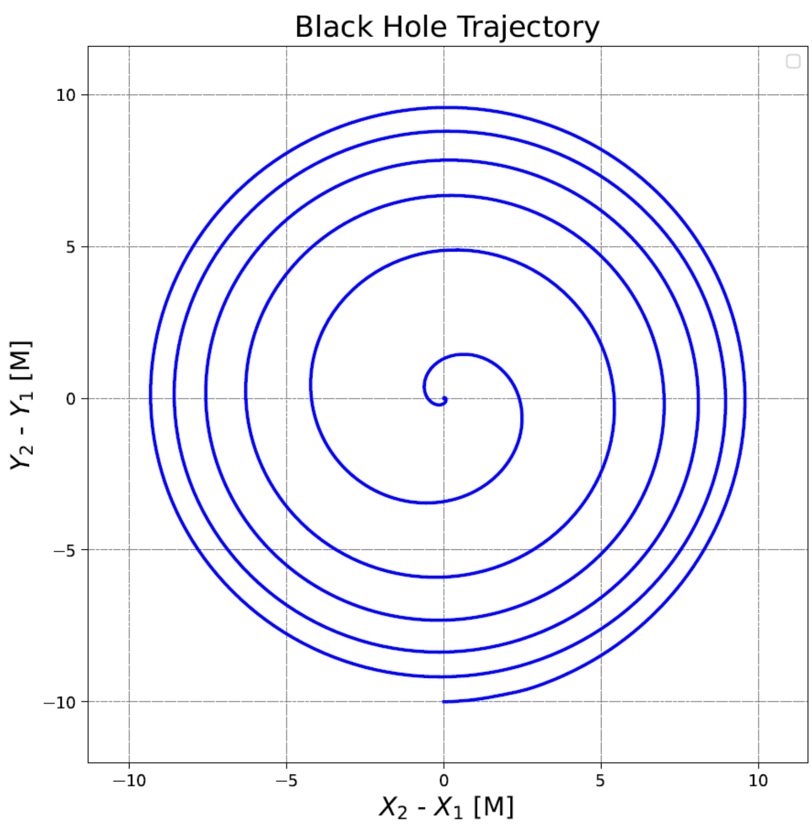
GW150914/

|___ AMSS_NCKU_source_copy

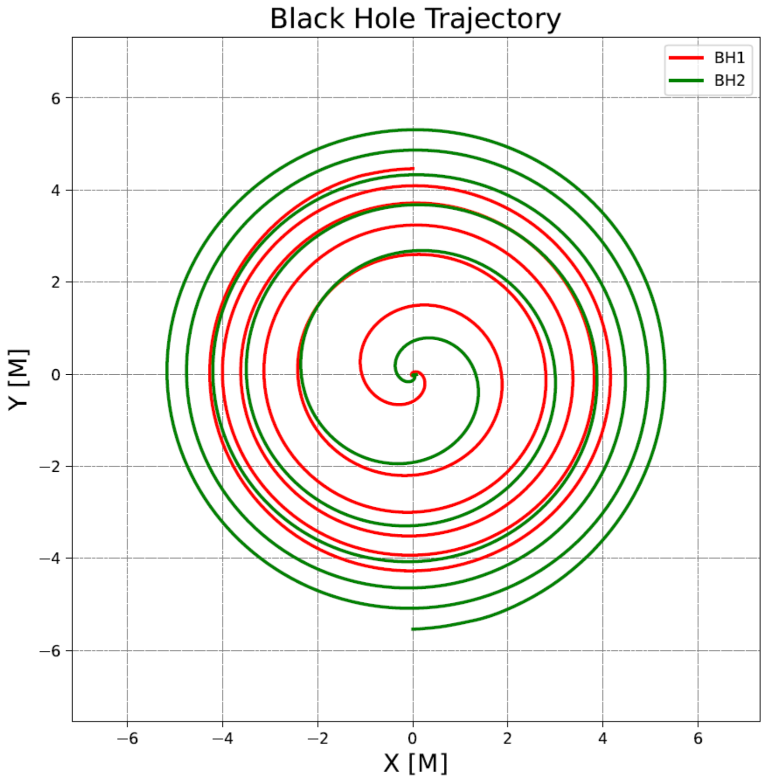
|___ AMSS_NCKU_output
|___ figure
    |___ **BH_Trajectory_XY.pdf**
    |___ **BH_Trajectory_21_XY.pdf**
    |___ **ADM_Constraint_Grid_Level_0.pdf**
    ......
|___ AMSS_NCKU_Input.py
|___ AMSS-NCKU.input
|___ AMSS-NCKU-TwoPuncture.input
|___ AMSS_NCKU_Resolution
|___ Initial_Grid.pdf
|___ Initial_Grid.jpeg
|___ macrodef.h
|___ macrodef.fh

The figures in **BH_Trajectory_XY.pdf** and **BH_Trajectory_21_XY.pdf** files reveal important information on black holes' trajectories during the in spiral and merger process. The figure **ADM_Constraint_Grid_Level_0.pdf** shows the **constraint violation** in the outermost grid level, which can quantify the accuracy in a numerical relativity simulation. When the simulation is evaluated over the physical time interval from T = 0 to T = 1000, the black hole trajectory exhibits the following characteristics:

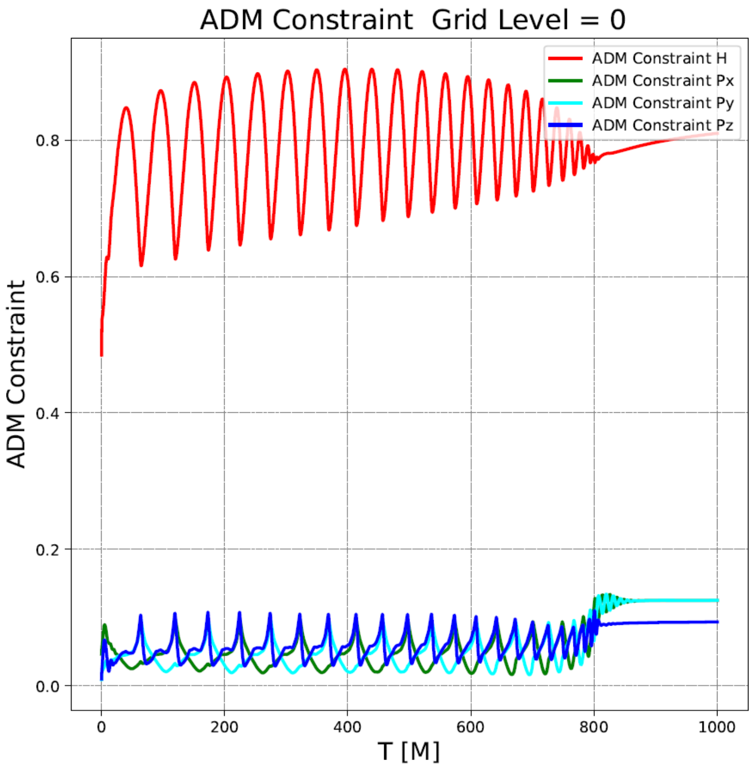
BH_Trajectory_21_XY.pdf:



BH_Trajectory_XY.pdf



ADM_Constraint_Grid_Level_0.pdf



Submission of the Results

After completing optimization and execution of the AMSS-NCKU binary black hole merger simulation, teams must compress the output directory “GW150914” and upload it to the designated FTP server as “GW150914.tar.gz”, as specified above. The directory must contain all required output files listed in the preceding sections.

Evaluation Criteria

The evaluation will focus on both the **correctness of the results** and the **degree of performance improvement** achieved. Proposals must include a detailed description of the optimization strategies employed, together with a clear explanation of the underlying technical principles. The following points outline key considerations for evaluation.

1. Participants must submit all required files in the compressed archive. Missing files will result in a score of zero for this part.
2. Participants must provide comprehensive documentation which will serve as the primary basis for scoring by the ASC26 Committee:
    - Installation process
    - Machine specifications
    - Optimization strategies
3. For the binary black hole trajectories, the optimized results must meet the following criteria:
    - **Accuracy Requirement**: The root mean square error (RMS error) *RMS*—reflecting the numerical deviation shown in BH_Trajectory_XY.pdf— must remain below **1%**.

      $$RMS = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\frac{|r_{1i}-r_{2i}|}{\max\left(r_{1i},r_{2i}\right)}\right)^2}$$

      r_{1i}, r_{2i} represent the coordinates of the black hole trajectories obtained from testing before and after optimization, respectively. And *M* denotes the integration timestep.
    - **Constraint Violation**: The file ADM_Constraint_Grid_Level_0.pdf presents the constraint violations on the outermost grid level. The absolute value of the constraint violation must not exceed 2, as this metric serves as a quantitative measure of the overall accuracy of the simulation.
    - **Required Submission**: The following three files must be included in the proposal:
        i. BH_Trajectory_XY.pdf
        ii. BH_Trajectory_21_XY.pdf
        iii. ADM_Constraint_Grid_Level_0.pdf

Failure to pass the validation test or to submit all specified files will result in a score of zero for this section.

Notes:

1. Any code that is related to the method parameters MUST NOT be modified.
2. Except for the number of MPI processors, all other input parameters must remain unchanged.
3. Please keep the directory structure and the resulting files unchanged.
4. The ASC organizing committee recommends using only the CPU version, as the GPU version has not been maintained for a long time and may contain unknown bugs.
5. If optimization is performed, the new algorithm must be mathematically equivalent to the original one.
6. Parameter-specific or case-specific optimizations are not permitted.
7. The proposal for this section should not exceed 30 pages.

Should you have any questions or need assistance, we warmly invite you to contact us at techsupport@asc-events.org. Our team is ready to support you throughout your ASC26 journey!

--End--