Chat history

ChatGPT
New chat

Yesterday
- Numerical Integration Methods

Previous 7 Days
- Sample Caps for Polished Microscopy
- More Time for Research
- Error in NumPy division.
- Brain Injury Assessment: Questions
- Letter of Recommendation Outline
- Gaussian Quadrature Function in Python
- STL File Composition
- Parallelize Python Code
- Check Python Version Mac
- Square Root in Python

Previous 30 Days
- Function Distribution Methods
- MPI Rank & Size
- Japanese 100 Phrases
- Unisex and Gender-Neutral
- Top 100 Japanese Nouns
- Quantifying Earth's Surface Roughness

February
- Generate and Export Matrices
- 3D Printing Overview
- Define Linux Environment File
- PhD Research Guidance
- Falstaff: Model Explanation
- Hazards in Work Environments
- Japan 2-Week Itinerary
- LinkedIn Message Group
- 20 Questions Game
- Clone Git Repo on HPC
- DFW Global Entry Locations

Upgrade planGet GPT-4, DALL·E, and more

James Smith
Close sidebar
ChatGPT 3.5

You
why am I getting this error in the code that will follow in the next message?
/apps/intel/oneapi/intelpython/latest/lib/python3.7/site-packages/numpy/polynomial/legendre.py:1581: RuntimeWarning: divide by zero encountered in true_divide
  w = 1/(fm * df)
/apps/intel/oneapi/intelpython/latest/lib/python3.7/site-packages/numpy/polynomial/legendre.py:1455: RuntimeWarning: divide by zero encountered in true_divide
  scl = 1./np.sqrt(2*np.arange(n) + 1)
Traceback (most recent call last):
  File "Homework7.py", line 45, in <module>
    comm.scatter(N, root=0)
  File "mpi4py/MPI/Comm.pyx", line 1267, in mpi4py.MPI.Comm.scatter
  File "mpi4py/MPI/msgpickle.pxi", line 730, in mpi4py.MPI.PyMPI_scatter
  File "mpi4py/MPI/msgpickle.pxi", line 119, in mpi4py.MPI.Pickle.dumpv
ChatGPT
ChatGPT
The error message you're encountering seems to be related to numerical calculations involving division by zero. Specifically, it's originating from the `numpy.polynomial.legendre` module.

Here are a few possibilities as to why you might be encountering this error:

Input Data Issue: Check the input data that you're passing to the function. It's possible that the data is causing division by zero.
Algorithm Issue: There could be a bug in the algorithm or function you're using that's leading to division by zero.
Boundary Issue: Division by zero often occurs when an algorithm or function encounters a boundary condition that isn't handled properly.
Library Version: It's possible that there's a bug in the version of NumPy or related libraries you're using. Updating to the latest version might solve the problem.

To address the issue:

- Double-check your input data to ensure it's valid and doesn't contain any zeros or other problematic values.
- Look into the algorithm or function you're using to see if there are any boundary conditions or special cases that need to be handled differently.
- Consider updating your NumPy or related libraries to the latest version to see if the issue has been addressed in newer releases.

Without seeing the specific code causing this issue, it's challenging to provide a more precise solution. If you could share the relevant part of your code, I might be able to provide more targeted assistance.

03

You

```
import math
from mpi4py import MPI
import numpy as np

# This code was written by James Smith to perform Gaussian Integration

# Start the MPI Process
comm = MPI.COMM_WORLD

# Determine the Total Number of Tasks
size = comm.Get_size()

# Determine the rank (ID) of this task
rank = comm.Get_rank()

# Function to integrate
def f(x):
    return x*math.exp(x)


N = list(range(1,21))


## --- Master ---
if rank == 0:

    # Generate Quadrature Number
    N = list(range(1,21))

    # Integration Bounds
    x1 = -1
    x2 = 1

    # Weights(w) and Integration Points(x)
    w=[]
    x=[]
    for i in N:
```

```
        x_temp,w_temp  = np.polynomial.legendre.leggauss(i)
        w.append(w_temp)
        x.append(x_temp)

    ### Compute the Quadrature ###

    # Send Data to Workers
    comm.scatter(N, root=0)
    comm.scatter(x, root=0)
    comm.scatter(w, root=0)

    # Collect the results from each worker
    int_results=[]
    for i in range(1,size):
        temp=comm.recv(source=i)
        int_results.append(temp)
    # Compute the table
    print(int_results)

## --- Worker ---
else:
    print(f"Process {rank} starts") #Displays worker ID

    # Receive Data
    N_local = comm.scatter(N, root=0)
    x_local = comm.scatter(x, root=0)
    w_local = comm.scatter(w, root=0)

    # Takes 3 arrays and isolates data list specific to this worker
    data_local=[N_local[rank-1],x_local[rank-1],w_local[rank-1]]

    # Takes Data and separates list into 3 arrays for weights, x, and N
    w_worker = data_local[2]
    x_worker = data_local[1]
    N_worker = data_local[0]

    # Performs Integration
    sum = 0.0
    for i in range (0,N_worker):
        sum = sum + w_worker[i]*f(x_worker[i])
    result=sum

    # Send Results to Main
    comm.send(result, dest=0)
```

Here is the code^

ChatGPT

ChatGPT

The error message you're encountering seems to originate from the `numpy.polynomial.legendre.leggauss` function. This function computes the Gauss-Legendre quadrature points and weights. The warning indicates that there might be division by zero happening within this function, which suggests that the legendre polynomial computation is encountering some numerical instability.

Given the nature of the legendre polynomial, division by zero can occur if the function is evaluated at certain points where the derivative becomes zero. This can happen when the number of quadrature points is low, or when the interval of integration is not properly specified.

To address this issue, you may consider the following:

- Increase Quadrature Points: Increase the number of quadrature points (`N`) to see if the error persists. More points might provide better accuracy and avoid division by zero issues.
- Check Integration Bounds: Ensure that the integration bounds (`x1` and `x2`) are properly specified. The legendre polynomial is defined on the interval [-1, 1], so if your bounds differ, you may encounter numerical instability.
- Handle Numerical Instability: If increasing quadrature points or adjusting bounds doesn't resolve the issue, you may need to handle numerical instability by catching the warning or using a different integration method.
- Debug Legendre Polynomial: Debug the legendre polynomial computation to understand where the division by zero is occurring. It might give you insights into what's causing the issue.

Try adjusting the code based on these suggestions and see if the error persists.

03

You

what is the proper syntax for numpy.polynomial.legendre.leggauss

ChatGPT

ChatGPT

The proper syntax for `numpy.polynomial.legendre.leggauss` is as follows:

python

Copy code

```
numpy.polynomial.legendre.leggauss(deg)
```

Where:

- `deg`: int, The degree of the legendre polynomial to use. It determines the number of quadrature points and weights.

This function returns two arrays:

- `x`: 1-D array, The roots (nodes) of the legendre polynomial of degree `deg`.
- `w`: 1-D array, The weights corresponding to the legendre polynomial of degree `deg`.

Here's an example of how you can use `numpy.polynomial.legendre.leggauss`:

python

Copy code

```
import numpy as np

# Degree of the legendre polynomial
degree = 5

# Compute Legendre-Gauss quadrature points and weights
x, w = np.polynomial.legendre.leggauss(degree)

print("Quadrature Points (Roots):", x)
print("Weights:", w)
```

This will compute the Legendre-Gauss quadrature points and weights for a Legendre polynomial of degree 5.

03

You
Assignment:
Write an MPI code with Python that computes the integral of $f(x) = xex$ using quadrature values of n = 1, 2, ... , 20. That is, 20 evaluations of the integral will be obtained, each with a different quadrature. Each integral is to be computed by a different worker process.
The Master computes the quadrature (x and w values), passes the quadrature to a worker

process, collects the results from each process, then constructs and prints a table with the results
of the form:
Quadrature no. Integration Result Percent error Run time (s)
1
2
3
...
20
The Python function gauleg.py can be called to obtain the quadrature for any number of points and for arbitrary integration limits. This code can be found in /work/ME5773/sec6/Homework.
The worker processes obtain the quadrature, evaluate the integral, and send the results back to the master process.
Part 1: write the file that is able to execute the MPI program, and run the program with 21 tasks (1 master, and 20 workers). Save the outputs of run in a text file named ‘part1.txt’
Part 2 (15% extra credit): On a different python file, modify the program for the case where the number of tasks is < 20 but still completes the table for up to 20 integration points. Write the outputs of this run in ‘part2.txt’.

ChatGPT

ChatGPT

To complete this assignment, you'll need to write two Python files: one for the MPI program itself and another for running the MPI program with different task counts. Below, I'll provide the code for both parts:

Part 1: MPI Program (mpi_integration.py)

python

Copy code

```
import math
from mpi4py import MPI
import numpy as np
import time

# Function to integrate
def f(x):
 return x * math.exp(x)

# Function to compute legendre-gauss quadrature points and weights
def compute_quadrature(n):
 x, w = np.polynomial.legendre.leggauss(n)
```

```
 return x, w

# Master process
def master(size):
 # Create table header
 print("Quadrature no.\tIntegration Result\tPercent error\tRun time (s)")

 # Loop over quadrature points
 for n in range(1, 21):
 # Start timer
 start_time = time.time()

 # Compute quadrature
 x, w = compute_quadrature(n)

 # Send quadrature to worker
 comm.send((x, w), dest=n, tag=1)

 # Receive result from worker
 result = comm.recv(source=n, tag=2)

 # Stop timer
 end_time = time.time()

 # Compute percent error
 exact_result = math.exp(1) - 1 # Exact integral of f(x) = x * exp(x) from -1 to 1
 percent_error = abs((result - exact_result) / exact_result) * 100

 # Print table row
 print(f"{n}\t\t{result}\t\t{percent_error:.6f}%\t\t{end_time - start_time:.6f}")

# Worker process
def worker(rank):
 # Receive quadrature from master
 x, w = comm.recv(source=0, tag=1)

 # Compute local integral
 local_result = sum(w[i] * f(x[i]) for i in range(len(x)))

 # Send result to master
 comm.send(local_result, dest=0, tag=2)

if __name__ == "__main__":
 comm = MPI.COMM_WORLD
 size = comm.Get_size()
```

```
 rank = comm.Get_rank()

 if size != 21:
 if rank == 0:
 print("Error: This program requires exactly 21 processes (1 master and 20
workers).")
 exit()

 if rank == 0:
 master(size)
 else:
 worker(rank)
```

Part 2: MPI Program with Fewer Tasks (mpi_integration_part2.py)

python

Copy code

```
import math
from mpi4py import MPI
import numpy as np
import time

# Function to integrate
def f(x):
 return x * math.exp(x)

# Function to compute legendre-gauss quadrature points and weights
def compute_quadrature(n):
 x, w = np.polynomial.legendre.leggauss(n)
 return x, w

# Master process
def master(size):
 # Create table header
 print("Quadrature no.\tIntegration Result\tPercent error\tRun time (s)")

 # Loop over quadrature points
 for n in range(1, size):
 # Start timer
 start_time = time.time()
```

```
 # Compute quadrature
 x, w = compute_quadrature(n)

 # Send quadrature to worker
 comm.send((x, w), dest=n, tag=1)

 # Receive result from worker
 result = comm.recv(source=n, tag=2)

 # Stop timer
 end_time = time.time()

 # Compute percent error
 exact_result = math.exp(1) - 1 # Exact integral of f(x) = x * exp(x) from -1 to 1
 percent_error = abs((result - exact_result) / exact_result) * 100

 # Print table row
 print(f"{n}\t\t{result}\t\t{percent_error:.6f}%\t\t{end_time - start_time:.6f}")

# Worker process
def worker(rank):
 # Receive quadrature from master
 x, w = comm.recv(source=0, tag=1)

 # Compute local integral
 local_result = sum(w[i] * f(x[i]) for i in range(len(x)))

 # Send result to master
 comm.send(local_result, dest=0, tag=2)

if __name__ == "__main__":
 comm = MPI.COMM_WORLD
 size = comm.Get_size()
 rank = comm.Get_rank()

 if rank == 0:
 master(size)
 else:
 worker(rank)
```

To run these scripts, you need to execute them using MPI. For example:

Copy code

```
mpirun -np 21 python mpi_integration.py > part1.txt
```

Copy code

```
mpirun -np 10 python mpi_integration_part2.py > part2.txt
```

These commands will execute the MPI programs with 21 and 10 processes respectively and save the outputs to `part1.txt` and `part2.txt`.

03

You
what is the syntax for the comm.scatter command
ChatGPT
ChatGPT
The syntax for the `comm.scatter` command in MPI4py is as follows:

python

Copy code

```
comm.scatter(sendobj, recvobj=None, root=0)
```

Where:

- `sendobj`: Object to be scattered. This can be a Python list, NumPy array, or any other iterable.
- `recvobj`: Buffer where the received data will be placed. This argument is required for non-root processes (workers) and can be omitted for the root process (master).
- `root`: Rank of the sending process (master) that scatters the data. This argument is optional and defaults to 0.

Here's an example of how `comm.scatter` can be used:

python

Copy code

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
size = comm.Get_size()
```

```
rank = comm.Get_rank()

# Data to be scattered
data = None
if rank == 0:
 data = [1, 2, 3, 4, 5]

# Scatter data to all processes
recv_data = comm.scatter(data, root=0)

print(f"Process {rank}: Received data = {recv_data}")
```

In this example, process 0 scatters the data `[1, 2, 3, 4, 5]` to all other processes using `comm.scatter`. Each process receives one element of the scattered data. The received data is stored in the `recv_data` variable.

03

You

ValueError Traceback (most recent call last)
File /work/ayn868/ME5773SP24/assignment7/Homework7.py:45
    40 x.append(x_temp)
    42 ### Compute the Quadrature ###
    43
    44 # Send Data to Workers
---> 45 comm.scatter(N, root=0)
    46 comm.scatter(x, root=0)
    47 comm.scatter(w, root=0)

File mpi4py/MPI/Comm.pyx:1587, in mpi4py.MPI.Comm.scatter()

File mpi4py/MPI/msgpickle.pxi:823, in mpi4py.MPI.PyMPI_scatter()

File mpi4py/MPI/msgpickle.pxi:161, in mpi4py.MPI.pickle_dumpv()

ValueError: expecting 1 items, got 20

ChatGPT

ChatGPT

The error message indicates a mismatch between the number of items being scattered and the number of processes in the communicator.

In MPI, when using `comm.scatter`, the number of items in the send buffer must match the number of processes in the communicator, except for the root process where the send buffer can be larger.

In your case, it seems like you're trying to scatter the list `N`, `x`, and `w` to all processes including the root process. However, you should only scatter to worker processes, not to the root process itself.

Here's how you can modify your code to fix this issue:

python

Copy code

```
## --- Master ---
if rank == 0:
 # Generate Quadrature Number
 N = list(range(1,21))

 # Integration Bounds
 x1 = -1
 x2 = 1

 # Weights(w) and Integration Points(x)
 w=[]
 x=[]
 for i in N:
 x_temp,w_temp = np.polynomial.legendre.leggauss(i)
 w.append(w_temp)
 x.append(x_temp)

 # Send Data to Workers
 comm.scatter(N[1:], root=0) # Send N to all workers except root
 comm.scatter(x[1:], root=0) # Send x to all workers except root
 comm.scatter(w[1:], root=0) # Send w to all workers except root
```

This modification sends `N`, `x`, and `w` to all worker processes (`rank > 0`), skipping the root process (`rank = 0`). This should resolve the `ValueError` you encountered.

03

You
will the brackets on N,x, and w cause them to only share part of the data?

ChatGPT

ChatGPT
Yes, you're correct. Using slicing with brackets (`[1:]`) will cause only a part of the data to be shared. This is because slicing extracts a portion of the list starting from the index 1 (the second element) to the end of the list.

If you want to send the entire lists `N`, `x`, and `w` to all worker processes except the root process, you should not use slicing. Instead, you should send the entire lists directly. Here's the corrected code:

python

Copy code

```
## --- Master ---
if rank == 0:
 # Generate Quadrature Number
 N = list(range(1,21))

 # Integration Bounds
 x1 = -1
 x2 = 1

 # Weights(w) and Integration Points(x)
 w=[]
 x=[]
 for i in N:
 x_temp,w_temp = np.polynomial.legendre.leggauss(i)
 w.append(w_temp)
 x.append(x_temp)

 # Send Data to Workers
 comm.scatter(N, root=0) # Send N to all workers including root
 comm.scatter(x, root=0) # Send x to all workers including root
 comm.scatter(w, root=0) # Send w to all workers including root
```

This modification sends the entire lists `N`, `x`, and `w` to all worker processes (including the root process). It ensures that each worker process receives the complete data for `N`, `x`, and `w`.

03

You

how do I specify to send to all workers except root for comm.scatter?

ChatGPT

ChatGPT

I apologize for the confusion. My previous responses were incorrect. The `comm.scatter` function in MPI doesn't have an option to specify excluding the root process. Instead, you should only scatter to the worker processes and not to the root process.

To achieve this, you can use an if condition to ensure that the root process doesn't scatter data to itself. Here's how you can modify your code:

python

Copy code

```
## --- Master ---
if rank == 0:
 # Generate Quadrature Number
 N = list(range(1,21))

 # Integration Bounds
 x1 = -1
 x2 = 1

 # Weights(w) and Integration Points(x)
 w=[]
 x=[]
 for i in N:
 x_temp,w_temp = np.polynomial.legendre.leggauss(i)
 w.append(w_temp)
 x.append(x_temp)

 # Send Data to Workers
 for i in range(1, size): # Skip root process (rank 0)
 comm.send(N, dest=i, tag=1)
 comm.send(x, dest=i, tag=2)
 comm.send(w, dest=i, tag=3)
```

This modification sends `N`, `x`, and `w` to all worker processes except the root process (`rank != 0`). The root process iterates over all worker processes and sends the data to each one individually using the `comm.send` function.

03

You
how do I receive this tagged data?

ChatGPT

ChatGPT
To receive tagged data in MPI4py, you can use the `recv` method with the appropriate tag. You'll need to ensure that each worker process receives the correct tagged data sent by the master process.

Here's how you can modify your worker process to receive tagged data:

python

Copy code

```
## --- Worker ---
else:
 print(f"Process {rank} starts") #Displays worker ID

 # Receive Data
 N_local = comm.recv(source=0, tag=1)
 x_local = comm.recv(source=0, tag=2)
 w_local = comm.recv(source=0, tag=3)

 # Perform computations using received data
 # ...

 # Send Results back to Main
 comm.send(result, dest=0, tag=4) # Adjust tag as needed
```

In this modification:

- Each worker process (except the root process) uses `comm.recv` to receive data from the root process (rank 0) with the specified tag.
- Each piece of data (N, x, and w) is tagged separately so that workers can receive them individually.

- After performing computations using the received data, the worker process can send the results back to the root process with an appropriate tag (e.g., tag=4).

Make sure to adjust the tag numbers according to your application's needs and ensure that they match the tags used in the master process for sending data.

03

You
can I send without a destination?

ChatGPT

ChatGPT

In MPI, when using the `comm.send` function in Python with MPI4py, you must specify a destination process to which the message will be sent. You cannot send a message without specifying a destination process.

Every communication in MPI involves a sender process and a receiver process. The `comm.send` function requires you to specify the destination process to which the message will be sent. This ensures that the message is received by the intended process.

Here's an example of how you can use `comm.send` to send data from one process to another:

python

Copy code

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0: # If the process is the sender
 data = "Hello, world!"
 destination = 1 # Destination process rank
 comm.send(data, dest=destination) # Sending data to destination process
elif rank == 1: # If the process is the receiver
 source = 0 # Source process rank
 received_data = comm.recv(source=source) # Receiving data from source process
 print("Received:", received_data)
```

In this example, process 0 sends the message "Hello, world!" to process 1 using `comm.send`, specifying process 1 as the destination (`dest=1`). Process 1 receives the message using `comm.recv`, specifying process 0 as the source (`source=0`).

03

You
what is comm.bcast
ChatGPT
ChatGPT
`comm.bcast` is a collective communication operation in MPI (Message Passing Interface) that broadcasts data from one process to all other processes in a communicator. It is used to distribute the same data from one process (often referred to as the "root" process) to all other processes in the communicator.

The syntax for `comm.bcast` in MPI4py is as follows:

python

Copy code

```
comm.bcast(sendobj, root=0)
```

Where:

- `sendobj`: The data object that will be broadcasted from the root process to all other processes. This can be any picklable Python object (e.g., a list, tuple, NumPy array, etc.).
- `root`: The rank of the root process that has the data to be broadcasted. This argument is optional and defaults to 0.

Here's an example of how `comm.bcast` can be used:

python

Copy code

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
```

```
if rank == 0:
 data = [1, 2, 3, 4, 5] # Data to be broadcasted
else:
 data = None # Placeholder for received data

data = comm.bcast(data, root=0) # Broadcasting data from root process to all other processes

print(f"Process {rank}: Received data: {data}")
```

In this example, process 0 has the data `[1, 2, 3, 4, 5]`, and it broadcasts this data to all other processes in the communicator using `comm.bcast`. All processes (including process 0) receive the broadcasted data.

03

You
how do I receive this broadcast
ChatGPT
ChatGPT
To receive a broadcasted message in MPI4py, you can use the `comm.bcast` function. This function will broadcast the data from the root process to all other processes in the communicator. Each process, including the root process, can receive the broadcasted data.

Here's how you can receive a broadcasted message:

python

Copy code

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()

if rank == 0:
 data = [1, 2, 3, 4, 5] # Data to be broadcasted
else:
 data = None # Placeholder for received data
```

```
data = comm.bcast(data, root=0) # Broadcasting data from root process to all other processes

print(f"Process {rank}: Received data: {data}")
```

In this example, process 0 has the data `[1, 2, 3, 4, 5]`, and it broadcasts this data to all other processes in the communicator using `comm.bcast`. All processes (including process 0) receive the broadcasted data.

Each process calls `comm.bcast`, specifying the data to be broadcasted and the root process (rank 0). The return value of `comm.bcast` is the broadcasted data, so you can assign it to a variable (`data` in this case) to receive the broadcasted message.

03

You
what does scattering do vs. broadcasting?

ChatGPT
ChatGPT
Scattering and broadcasting are both collective communication operations in MPI, but they serve different purposes and have different behaviors:

Scattering:

- Scattering is a collective communication operation where the root process sends different pieces of data to different processes in the communicator.
- The data to be scattered is divided into segments, and each segment is sent to a different process.
- Each process receives a different segment of the data, and the segments may have different sizes.
- Scattering is useful when you have a large dataset that needs to be divided among different processes for parallel processing.

Broadcasting:

- Broadcasting is a collective communication operation where the root process sends the same piece of data to all other processes in the communicator.
- The root process has the data, and it sends copies of this data to all other processes.
- All processes in the communicator receive the same data.
- Broadcasting is useful when you have a piece of data that needs to be shared among all processes in the communicator.

In summary:

- Scattering divides data into segments and sends different segments to different processes.
- Broadcasting sends the same data from the root process to all other processes.

03

You
can you please explain this slurm job script #!/bin/bash
#----------
Sample Slurm job script for Arc nodes
#SBATCH -J myjob # Job Name
#SBATCH -o myjob.o%j # Name of the stdout output file
#SBATCH -e myjob.e%j # Name of stderr error file
#SBATCH -p compute1 # Queue (partition) name
#SBATCH -N 2 # Total # of nodes (must be 1 for serial)
#SBATCH -n 40 # Total # of mpi tasks (should be 1 for serial)
#SBATCH -t 00:10:00 # Run time (hh:mm:ss)
#SBATCH --mail-type=ALL
#SBATCH --mail-user=pratik.mitra@my.utsa.edu

Load module
module load openmpi
module swap openmpi intel/oneapi/2021.2.0
#module load libctl/4.5.0
#module load gcc
#module swap intel/oneapi gcc/11.2.0

cd /work/gug303/sources/ME5773SP24/assignment7

#mpirun -np 80 Homework7.py
mpiexec -n 80 python Homework7.py

ChatGPT

ChatGPT
Certainly! This is a Slurm job script, which is a shell script used to submit jobs to a high-performance computing (HPC) cluster managed by the Slurm workload manager. Let's break down the script and understand each part:

bash

Copy code

```
#!/bin/bash
```

This line specifies the interpreter to use, which is Bash in this case. It indicates that the script is written in the Bash scripting language.

bash

Copy code

```
#SBATCH -J myjob
```

This line sets the name of the job to `myjob`. Jobs submitted to the cluster will be identified by this name.

bash

Copy code

```
#SBATCH -o myjob.o%j
#SBATCH -e myjob.e%j
```

These lines specify the names of the output and error files for the job. `%j` is a Slurm job ID placeholder, which will be replaced by the actual job ID when the job is submitted.

bash

Copy code

```
#SBATCH -p compute1
```

This line specifies the partition or queue name where the job will be submitted. The job will be scheduled and run on the nodes allocated to this partition.

bash

Copy code

```
#SBATCH -N 2
```

This line specifies the number of nodes requested for the job. In this case, the job requests 2 nodes.

bash

Copy code

```
#SBATCH -n 40
```

This line specifies the total number of MPI tasks requested for the job. MPI tasks are typically processes that communicate with each other using the Message Passing Interface (MPI) protocol. In this case, the job requests 40 MPI tasks.

bash

Copy code

```
#SBATCH -t 00:10:00
```

This line specifies the maximum run time for the job. It is expressed in hours, minutes, and seconds (hh:mm:ss). In this case, the job is allowed to run for a maximum of 10 minutes.

bash

Copy code

```
#SBATCH --mail-type=ALL
#SBATCH --mail-user=pratik.mitra@my.utsa.edu
```

These lines specify that email notifications should be sent for job events of all types (`--mail-type=ALL`) to the specified email address (`pratik.mitra@my.utsa.edu`).

bash

Copy code

```
module load openmpi
module swap openmpi intel/oneapi/2021.2.0
```

These lines load specific software modules. The `module load` command loads the Open MPI library, and the `module swap` command swaps the previously loaded Open MPI module with a different version (`intel/oneapi/2021.2.0`).

bash

Copy code

```
cd /work/gug303/sources/ME5773SP24/assignment7
```

This line changes the current working directory to `/work/gug303/sources/ME5773SP24/assignment7`. This is where the script expects to find the Python script (`Homework7.py`) that will be executed.

bash

Copy code

```
#mpirun -np 80 Homework7.py
mpiexec -n 80 python Homework7.py
```

These lines are commented out, but they are options for running the MPI program. The first option (`mpirun`) is commented out, while the second option (`mpiexec`) is not. Both commands execute the Python script `Homework7.py` with 80 MPI tasks.

Overall, this Slurm job script specifies the job's characteristics, such as its name, requested resources (nodes, MPI tasks, run time), email notification settings, and software environment. It also sets up the execution environment and launches the MPI program with the specified parameters.

03

You
if I am trying to run the previous code we discussed with 20 workers + the master while using the minimum number of resources, what should I change in this file?

ChatGPT

ChatGPT
To run the previous code with 20 workers (plus the master process) while using the minimum number of resources, you would need to make the following changes to the Slurm job script:

- Reduce the number of nodes: Since you only need 21 processes (20 workers + 1 master), you can reduce the number of nodes to 1.
- Reduce the number of MPI tasks: Set the total number of MPI tasks to 21.

Adjust the run time: If the job will complete quickly, you can reduce the run time accordingly.

Here's the modified Slurm job script with these changes:

bash

Copy code

```
#!/bin/bash
#SBATCH -J myjob
#SBATCH -o myjob.o%j
#SBATCH -e myjob.e%j
#SBATCH -p compute1
#SBATCH -N 1 # Reduce the number of nodes to 1
#SBATCH -n 21 # Set the total number of MPI tasks to 21 (20 workers + 1 master)
#SBATCH -t 00:05:00 # Adjust the run time as needed
#SBATCH --mail-type=ALL
#SBATCH --mail-user=pratik.mitra@my.utsa.edu

module load openmpi
module swap openmpi intel/oneapi/2021.2.0

cd /work/gug303/sources/ME5773SP24/assignment7

mpiexec -n 21 python Homework7.py # Run the MPI program with 21 processes
```

With these modifications, the job script will request one node with a total of 21 MPI tasks (20 workers and 1 master). Adjust the run time as needed based on the expected execution time of your MPI program.

03

You
how do I show line numbers in vIM

ChatGPT

ChatGPT
To show line numbers in Vim, you can use the `:set number` command. This will display line numbers along the left side of the Vim editor window.

Here's how you can enable line numbers in Vim:

Open Vim by typing `vim` followed by the filename you want to edit.

Copy code

```
vim filename
```

Once inside Vim, press the Esc key to ensure you're in normal mode.
Type the following command and press Enter:

arduino

Copy code

```
:set number
```

This command tells Vim to display line numbers.
If you want Vim to remember this setting for future editing sessions, you can add the command to your `.vimrc` file. Open your `.vimrc` file (or create it if it doesn't exist) and add the following line:

arduino

Copy code

```
set number
```

Save the `.vimrc` file and restart Vim for the changes to take effect.

That's it! Now Vim will display line numbers in the editor window, making it easier to navigate through your files.

03

You
when I run my code, I get an error saying that the x variable is undefined in the x_local = comm.bcast(x,root=0) line

ChatGPT

ChatGPT
The error you're encountering suggests that the variable `x` is undefined at the point where you're attempting to use it as an argument for `comm.bcast`.

To resolve this issue, you need to ensure that the variable `x` is defined and initialized before you attempt to broadcast it using `comm.bcast`. Here's a general approach to ensure the variable `x` is defined:

> Define and initialize the variable `x`: Ensure that `x` is defined and assigned a value before attempting to broadcast it. This could involve computing the value of `x` or loading it from an external source.
> Check the scope of `x`: Make sure that `x` is defined within the scope of the process where `comm.bcast` is called. If `x` is defined in a conditional block or function that is not executed in the process where `comm.bcast` is called, it will be undefined.

Here's an example of how you can define and initialize `x` before broadcasting it:

python

Copy code

```
# Master process
if rank == 0:
 # Generate x
 x = generate_x()
else:
 x = None

# Broadcast x to all processes
x = comm.bcast(x, root=0)
```

In this example:

- In the master process (rank 0), `x` is generated using a function `generate_x()`.
- In all other processes, `x` is initialized as `None`.
- Then, `comm.bcast` is used to broadcast `x` from the master process to all other processes.

Ensure that `generate_x()` or the method you're using to define `x` is properly implemented and that it returns a valid value. Additionally, make sure that the variable `x` is defined in the scope where `comm.bcast` is called.

03

You

how would I receive this x broadcast in the worker?

ChatGPT

ChatGPT

To receive the broadcasted variable `x` in the worker process, you can use the `comm.bcast` function. In MPI4py, all processes in the communicator call `comm.bcast`, specifying the variable to be broadcasted (`x`) and the root process (in this case, root=0).

Here's how you can receive the broadcasted variable `x` in the worker process:

python

Copy code

```
# Worker processes
else:
 # Receive the broadcasted variable x
 x = comm.bcast(None, root=0)

 # Now x is available in all worker processes
 # Proceed with your computations using x
```

In this code snippet:

- In the worker process (where rank != 0), the `comm.bcast` function is called to receive the broadcasted variable `x`.
- The `comm.bcast` function is called with `None` as the argument for the variable to be broadcasted. This is because the root process (rank=0) specifies the variable to be broadcasted, and other processes only need to specify the root process.
- After calling `comm.bcast`, the variable `x` will contain the broadcasted value in all worker processes, and you can proceed with your computations using `x`.

Make sure to place this code snippet in the worker process block of your MPI code where you need to use the broadcasted variable `x`.

03

You

if I am broadcasting multiple lists at once, how do I specify which list is received?

ChatGPT

ChatGPT

When broadcasting multiple lists at once in MPI4py, you can use tags to distinguish between the different lists being broadcasted and received. Each list can be assigned a unique tag, allowing processes to identify which list they are receiving.

Here's how you can broadcast multiple lists with different tags and receive them in worker processes:

Master process:

python

Copy code

```
# Master process (rank 0)
if rank == 0:
 # Broadcast list 'x' with tag 1
 comm.bcast(x, root=0, tag=1)
 # Broadcast list 'y' with tag 2
 comm.bcast(y, root=0, tag=2)
 # Broadcast list 'z' with tag 3
 comm.bcast(z, root=0, tag=3)
```

Worker processes:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive list 'x' with tag 1
 x_received = comm.bcast(None, root=0, tag=1)
 # Receive list 'y' with tag 2
 y_received = comm.bcast(None, root=0, tag=2)
 # Receive list 'z' with tag 3
 z_received = comm.bcast(None, root=0, tag=3)
```

In this example:

- The master process broadcasts three lists (`x`, `y`, and `z`) to all other processes.

- Each list is broadcasted with a unique tag (`1`, `2`, and `3` respectively).
- In the worker processes, the `comm.bcast` function is called to receive each list with the corresponding tag.
- After calling `comm.bcast`, the received lists (`x_received`, `y_received`, and `z_received`) contain the broadcasted lists in the worker processes.

By using different tags for each broadcast and receive operation, you can ensure that the worker processes correctly identify and receive each list.

03

You
what is the significance of the None in the first field of the receiving broadcast

ChatGPT

ChatGPT
In MPI4py, the `comm.bcast` function is used for broadcasting data from one process to all other processes in the communicator. When receiving a broadcast in a worker process, you might notice that `None` is passed as the first argument to `comm.bcast`. Let's discuss the significance of passing `None` in this context:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive list 'x' with tag 1
 x_received = comm.bcast(None, root=0, tag=1)
```

In this code snippet:

- The `comm.bcast` function is being called in the worker process to receive the broadcasted data.
- The first argument (`None`) in the `comm.bcast` function call is used as a placeholder. It is not used to transmit any actual data.
- The purpose of passing `None` as the first argument is to indicate to MPI4py that the process is a receiver of the broadcast and to specify additional parameters such as the source root and the tag.
- When `comm.bcast` is called with `None` as the first argument, MPI4py expects to receive data from the root process specified by the `root` parameter and identified by the `tag` parameter.

In summary, passing `None` as the first argument in the `comm.bcast` call when receiving a broadcast in MPI4py indicates to the MPI library that the process is a receiver of the broadcast and expects to receive data from the specified root process. The actual data being broadcasted is transmitted by the root process, and `None` serves as a placeholder to trigger the receive operation.

03

You
when would i put anything other than None in that field?

ChatGPT

ChatGPT
In MPI4py, the first argument of the `comm.bcast` function when receiving a broadcast is typically set to `None` because it acts as a placeholder. However, there are situations where you might put something other than `None` in that field:

Sending Data: If the worker process is also broadcasting data to other processes, you would put the data you want to broadcast in place of `None`. For example:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Broadcast data_list to all processes
 data_list = [1, 2, 3, 4, 5]
 comm.bcast(data_list, root=0, tag=1)
```

In this case, `data_list` contains the data that the worker process wants to broadcast to all other processes. The `comm.bcast` function is used to broadcast `data_list` to all processes in the communicator.

Receiving Non-Root Process Data: If a worker process is receiving data broadcasted by another non-root process, you would put the received data type in place of `None`. For example:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive data_list broadcasted by process with rank 2
 data_list = comm.bcast([], root=2, tag=1)
```

> In this case, the worker process expects to receive data broadcasted by the process with rank 2. The empty list `[]` serves as a placeholder to trigger the receive operation. The actual broadcasted data will be received and stored in the `data_list` variable.

In summary, you would put something other than `None` in the first argument of `comm.bcast` when the worker process needs to either send data or receive data broadcasted by another non-root process. Otherwise, when receiving a broadcast from the root process, `None` is typically used as a placeholder.

03

You

Traceback (most recent call last):
  File "Homework7.py", line 62, in <module>
    N_local = comm.bcast(None, root=0, tag=1)
  File "mpi4py/MPI/Comm.pyx", line 1254, in mpi4py.MPI.Comm.bcast
TypeError: bcast() got an unexpected keyword argument 'tag'

ChatGPT

ChatGPT

The error indicates that the `comm.bcast` function does not expect the `tag` keyword argument. This is because the `tag` argument is not a valid parameter for the `bcast` function in MPI4py.

In MPI, communication functions such as `bcast` do not have a `tag` argument. Tags are used with point-to-point communication functions like `send` and `recv`, but not with collective communication functions like `bcast`, `scatter`, or `gather`.

Here's how you can modify your code to remove the `tag` argument:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive broadcasted data from root process
 N_local = comm.bcast(None, root=0)
```

In this modified code, the `comm.bcast` function is called without the `tag` argument. The worker process simply waits to receive the broadcasted data from the root process specified by the `root` argument (in this case, root=0). Make sure that the root process broadcasts the data without specifying a tag as well.

If you need to distinguish between different broadcasts in your application, you may consider using different `bcast` calls for each piece of data, each with its own unique tag, and corresponding `recv` calls in the worker processes. However, this is only applicable to point-to-point communication, not to collective communication functions like `bcast`.

03

You
why the fuck did you tell me to put tag?

ChatGPT

ChatGPT
I apologize for the confusion. It seems there was a misunderstanding. I provided incorrect information in my response.

In MPI4py, the `comm.bcast` function does not accept a `tag` argument. Tags are used with point-to-point communication functions like `send` and `recv`, not with collective communication functions like `bcast`.

To correct the issue, simply remove the `tag` argument from the `comm.bcast` function call. Here's the corrected code:

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive broadcasted data from root process
 N_local = comm.bcast(None, root=0)
```

I apologize for any inconvenience caused by the incorrect information. If you have any further questions or need clarification, please feel free to ask.

03

You
I want to broadcast 3 sets of data at once, how can I do this without mixing them up?

ChatGPT

ChatGPT
To broadcast multiple sets of data at once without mixing them up, you can use a simple approach where you broadcast each set of data separately. This ensures that each set of data is sent and received independently of the others. Here's how you can do it:

python

Copy code

```
# Master process (rank 0)
if rank == 0:
 # Broadcast each set of data separately
 comm.bcast(data1, root=0)
 comm.bcast(data2, root=0)
 comm.bcast(data3, root=0)
```

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive each set of data separately
 data1_received = comm.bcast(None, root=0)
 data2_received = comm.bcast(None, root=0)
 data3_received = comm.bcast(None, root=0)
```

In this approach:

- The master process broadcasts each set of data (`data1`, `data2`, `data3`) separately using `comm.bcast`.
- Each worker process receives each set of data separately using `comm.bcast`.
- Each set of data is sent and received independently, ensuring that they are not mixed up.

Make sure that `data1`, `data2`, and `data3` are defined and initialized before calling `comm.bcast`, and that they are properly received in the worker processes. This approach keeps the broadcast and receive operations separate, avoiding any mixing up of the data sets.

03

You
how does the worker know which order to receive the data in?

ChatGPT

ChatGPT
In the approach outlined above, the worker processes receive the data sets in the same order in which they are broadcasted by the master process. This order is determined by the sequence of `comm.bcast` calls made by the master process.

Let's consider the example provided earlier:

python

Copy code

```
# Master process (rank 0)
if rank == 0:
 # Broadcast each set of data separately
 comm.bcast(data1, root=0)
 comm.bcast(data2, root=0)
 comm.bcast(data3, root=0)
```

In this code snippet, `data1` is broadcasted first, followed by `data2`, and then `data3`.

python

Copy code

```
# Worker processes (rank != 0)
else:
 # Receive each set of data separately
 data1_received = comm.bcast(None, root=0)
 data2_received = comm.bcast(None, root=0)
 data3_received = comm.bcast(None, root=0)
```

In the worker processes, `data1` is received first, followed by `data2`, and then `data3`. This corresponds to the order in which they were broadcasted by the master process.

The order of broadcast and receive operations should match to ensure that the worker processes receive the data sets correctly. If the order of broadcast and receive operations needs to be different, you would need to adjust the code accordingly. For example, you could modify the master process to broadcast the data sets in a different order, or you could use tags to distinguish between different sets of data.

03

You
/apps/intel/oneapi/intelpython/latest/lib/python3.7/site-packages/numpy/polynomial/legendre.py:1455: RuntimeWarning: divide by zero encountered in true_divide
  scl = 1./np.sqrt(2*np.arange(n) + 1)
/apps/intel/oneapi/intelpython/latest/lib/python3.7/site-packages/numpy/polynomial/legendre.py:1581: RuntimeWarning: divide by zero encountered in true_divide
  w = 1/(fm * df)

ChatGPT

ChatGPT
These warnings are coming from the NumPy library, specifically from the `numpy.polynomial.legendre` module. They indicate that during the computation of Legendre polynomials, a division by zero was encountered.

The warning `RuntimeWarning: divide by zero encountered in true_divide` suggests that there is division by zero occurring in the computation. This often happens when trying to calculate the Legendre polynomials or associated values with an inappropriate input.

To handle these warnings, you can consider the following steps:

Check your input: Ensure that the input values you are passing to the Legendre polynomial functions are appropriate. Division by zero can occur when using invalid input values, such as when the degree of the polynomial or the range of values is not suitable.
Handle special cases: If you expect division by zero to occur in certain cases and you want to handle it gracefully, you can use try-except blocks to catch the warning and handle it appropriately in your code.
Suppress the warning (if appropriate): If you are confident that the division by zero is expected behavior in your specific use case and you do not want to see the warning, you

can suppress it using the `warnings` module in Python. However, be cautious when suppressing warnings, as they can provide valuable information about potential issues in your code.

Here's an example of how you can suppress the warning:

python

Copy code

```
import numpy as np
import warnings

# Suppress the warning
warnings.filterwarnings("ignore", category=RuntimeWarning)

# Your code that produces the warning
# For example:
# result = np.polynomial.legendre.leggauss(10)
```

Remember to handle warnings appropriately based on your specific use case and whether the warnings indicate potential issues in your code.

03

You

How do I pair a worker number with the data returned to the main?

ChatGPT

ChatGPT

To pair a worker number with the data returned to the main process, you can use a dictionary or a list to store the data along with the corresponding worker number. Here's a general approach:

Worker Processes:

- Each worker process computes some data.
- After computation, each worker process sends its data along with its worker number (rank) to the main process.

Main Process:

- Collects the data and worker numbers from all worker processes.
- Stores the data and worker numbers in a suitable data structure (such as a dictionary or a list).

Here's an example implementation using a dictionary to pair worker numbers with the data:

python

Copy code

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

# Worker processes
if rank != 0:
 # Simulated data computed by each worker process
 data = rank * 10 # Example: Worker number multiplied by 10

 # Send data and worker number to main process
 comm.send((rank, data), dest=0)

# Main process
else:
 # Dictionary to store data paired with worker numbers
 data_pairs = {}

 # Receive data from each worker process and store it in the dictionary
 for worker_rank in range(1, size):
 worker_data = comm.recv(source=worker_rank)
 worker_number, worker_data = worker_data
 data_pairs[worker_number] = worker_data

 # Print the paired data
 print("Data paired with worker numbers:")
 for worker_number, worker_data in data_pairs.items():
 print(f"Worker {worker_number}: {worker_data}")
```

In this example:

- Each worker process computes some data (in this case, a simulated data value based on its rank).
- After computation, each worker process sends its data along with its rank to the main process using `comm.send`.
- The main process receives the data from each worker process using `comm.recv` and stores it in a dictionary, pairing it with the corresponding worker number (rank).
- Finally, the main process prints the paired data.

You can modify this example to suit your specific use case and data types. Depending on your requirements, you might choose a different data structure or communication method.

03

You
how do I sort a list of tuples (paired worker number with return value) by the worker number

ChatGPT

ChatGPT
To sort a list of tuples containing paired worker numbers with their return values by the worker number, you can use the `sorted()` function in Python. You need to specify the key by which you want to sort the list. In this case, you want to sort by the first element of each tuple, which represents the worker number.

Here's how you can do it:

python

Copy code

```
# List of tuples containing paired worker numbers with return values
paired_data = [(2, 20), (1, 10), (3, 30)]

# Sort the list of tuples by the worker number (first element of each tuple)
sorted_paired_data = sorted(paired_data, key=lambda x: x[0])

# Print the sorted list
print("Sorted paired data by worker number:")
for worker_number, return_value in sorted_paired_data:
 print(f"Worker {worker_number}: {return_value}")
```

Output:

yaml

Copy code

```
Sorted paired data by worker number:
Worker 1: 10
Worker 2: 20
Worker 3: 30
```

In this example:

- `sorted()` function sorts the list of tuples (`paired_data`) based on the worker number (first element of each tuple).
- The `key` parameter specifies the function to be called on each element for sorting. Here, we use a lambda function to access the first element (`x[0]`) of each tuple.
- The sorted list (`sorted_paired_data`) is printed, displaying the worker numbers and their corresponding return values in sorted order.

03

You

```
## --- Worker ---
else:
    print(f"Process {rank} starts") #Displays worker ID

    # Receive Data
    N_local = comm.bcast(None, root=0)
    x_local = comm.bcast(None, root=0)
    w_local = comm.bcast(None, root=0)

    # Takes 3 arrays and isolates data list specific to this worker
    data_local=[N_local[rank-1],x_local[rank-1],w_local[rank-1]]

    # Takes Data and separates list into 3 arrays for weights, x, and N
    w_worker = data_local[2]
    x_worker = data_local[1]
    N_worker = data_local[0]

    # Performs Integration
    sum = 0.0
    for i in range (0,N_worker):
        sum = sum + w_worker[i]*f(x_worker[i])
    result=sum

    # Send Results to Main
    comm.send((rank, result), dest=0 Why does this return 0 for worker 1?
```

ChatGPT

ChatGPT

The code you provided seems to be part of an MPI program where worker processes are performing some computation and then sending the results back to the main process. Let's address the issue you mentioned, where worker 1 returns 0:

The likely reason why worker 1 returns 0 could be due to an indexing issue in your code when accessing the local data (`N_local`, `x_local`, `w_local`). In Python, indexing starts from 0, so accessing `N_local[rank-1]`, `x_local[rank-1]`, and `w_local[rank-1]` might be incorrect, especially if `rank` is 1-based.

Assuming `N_local`, `x_local`, and `w_local` are lists of data broadcasted from the main process, and `rank` is the worker process's rank, you should access the data using `rank` directly, without subtracting 1.

Here's how you should modify the code:

python

Copy code

```
else:
 print(f"Process {rank} starts") # Display worker ID

 # Receive Data
 N_local = comm.bcast(None, root=0)
 x_local = comm.bcast(None, root=0)
 w_local = comm.bcast(None, root=0)

 # Access local data directly using rank
 N_worker = N_local[rank]
 x_worker = x_local[rank]
 w_worker = w_local[rank]

 # Performs Integration
 result = 0.0
 for i in range(N_worker):
 result += w_worker[i] * f(x_worker[i])

 # Send Results to Main
 comm.send((rank, result), dest=0)
```

In this modified version:

- We access `N_local`, `x_local`, and `w_local` using the worker process's rank directly.

- The loop for integration starts from 0 up to `N_worker - 1`, where `N_worker` is the number of integration points for the specific worker process.
- Each worker process sends its rank along with the computed result to the main process.

03

You

Here is the master process Last login: Thu Mar 28 12:37:49 on ttys000
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
(ayn868@arc.utsa.edu) Password:
(ayn868@arc.utsa.edu) Duo two-factor login for ayn868

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-8036
2. Phone call to XXX-XXX-8036
3. SMS passcodes to XXX-XXX-8036

Passcode or option (1-3): 1
Success. Logging you in...
Last login: Thu Mar 28 12:38:26 2024 from 129.222.77.154

Welcome to Arc!

Do *not* execute code directly on the login nodes. SBATCH should be used to submit jobs to the Slurm job scheduling engine.

If you need to run an interactive job on a compute node, use the following syntax to connect to one:

```
srun -p compute1 -n 1 -t 00:05:00 --cpus-per-task=1 --pty bash

Starting in work directory /work/ayn868
[login001: ayn868]$ ls
HDFView  ME5773SP24  sec3  source_code  sources
[login001: ayn868]$ cd ME5773SP24/
[login001: ME5773SP24]$ cd assignment7
[login001: assignment7]$ ls
gauleg.py     job_script.slurm  myjob.e364679  myjob.o364679  part2.py
Homework7.py  myjob.e364675     myjob.o364675  part1.py       part2.txt
int-trap.py   myjob.e364678     myjob.o364678  part1.txt
[login001: assignment7]$ ml vim
The vim module version 8.1  is loaded.
[login001: assignment7]$ vim job_script.slurm
[login001: assignment7]$ vim part1.py
[login001: assignment7]$ vim
[login001: assignment7]$ ls
gauleg.py     int-trap.py       myjob.e364675  myjob.e364679  myjob.o364678  part1.py   part2.py
Homework7.py  job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679  part1.txt
part2.txt
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ ls
gauleg.py     int-trap.py       myjob.e364675  myjob.e364679  myjob.o364678  part1.py   part2.py
Homework7.py  job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679  part1.txt
part2.txt
[login001: assignment7]$ client_loop: send disconnect: Broken pipe
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
```

@@@
@@@@@@@@@@@@@@@@@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@
@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
@@@
@@@@@@@@@@@@@@@@@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@
@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
@@@
@@@@@@@@@@@@@@@@@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@
@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.

Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
@@@
@   WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!   @
@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:fDPYc5lZ38NmQ2Icyk2qmytzbxsnAOKBmt/H9anJtjo.
Please contact your system administrator.
Add correct host key in /Users/james/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /Users/james/.ssh/known_hosts:3
Host key for arc.utsa.edu has changed and you have requested strict checking.
Host key verification failed.
(base) james@Jamess-MacBook-Pro-2 ~ % ssh ayn868@arc.utsa.edu
(ayn868@arc.utsa.edu) Password:
(ayn868@arc.utsa.edu) Duo two-factor login for ayn868

Enter a passcode or select one of the following options:

1. Duo Push to XXX-XXX-8036
2. Phone call to XXX-XXX-8036
3. SMS passcodes to XXX-XXX-8036

Passcode or option (1-3): 1
Success. Logging you in...
Last login: Fri Mar 29 13:42:18 2024 from 129.222.77.154

Welcome to Arc!

Do *not* execute code directly on the login nodes. SBATCH should be used to submit jobs to the Slurm job scheduling engine.

If you need to run an interactive job on a compute node, use the following syntax to connect to one:

srun -p compute1 -n 1 -t 00:05:00 --cpus-per-task=1 --pty bash

Starting in work directory /work/ayn868

```
[login001: ayn868]$ ls
HDFView  ME5773SP24  sec3  source_code  sources
[login001: ayn868]$ cd ME5773SP24/
[login001: ME5773SP24]$ ls
assignment3  assignment4  assignment5  assignment6  assignment7  README.md
[login001: ME5773SP24]$ cd assignment7
[login001: assignment7]$ ls
gauleg.py    int-trap.py      myjob.e364675  myjob.e364679  myjob.o364678  part1.py   part2.py
Homework7.py  job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679  part1.txt
part2.txt
[login001: assignment7]$ vim Homework7.py
-bash: vim: command not found
[login001: assignment7]$ ml vim
The vim module version 8.1  is loaded.
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ ls
gauleg.py    int-trap.py      myjob.e364675  myjob.e364679  myjob.o364678  part1.py   part2.py
Homework7.py  job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679  part1.txt
part2.txt
[login001: assignment7]$ vim job_script.slurm
[login001: assignment7]$ cp job_script.slurm James_script.slurm
[login001: assignment7]$ vim James_script.slurm
[login001: assignment7]$ ls
gauleg.py    int-trap.py      job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679
part1.txt  part2.txt
Homework7.py  James_script.slurm  myjob.e364675    myjob.e364679  myjob.o364678
part1.py     part2.py
[login001: assignment7]$ sbatch James_script.slurm
Submitted batch job 365235
[login001: assignment7]$ squeue -u ayn868
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[login001: assignment7]$ ls
gauleg.py    Jamesjob.e365235   job_script.slurm  myjob.e364679  myjob.o364679  part2.py
Homework7.py  Jamesjob.o365235   myjob.e364675    myjob.o364675  part1.py     part2.txt
int-trap.py   James_script.slurm  myjob.e364678    myjob.o364678  part1.txt
[login001: assignment7]$ vim Jamesjob.e365235
[login001: assignment7]$ rm Jamesjob.
Jamesjob.e365235  Jamesjob.o365235
[login001: assignment7]$ rm Jamesjob.*
Do you really want to run this delete command?
 rm Jamesjob.e365235 Jamesjob.o365235
(y/n)? y
[login001: assignment7]$ ls
```

```
gauleg.py     int-trap.py        job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679
part1.txt  part2.txt
Homework7.py  James_script.slurm  myjob.e364675     myjob.e364679  myjob.o364678
part1.py       part2.py
[login001: assignment7]$ vim James_script.slurm
[login001: assignment7]$ pwd
/work/ayn868/ME5773SP24/assignment7
[login001: assignment7]$ vim James_script.slurm
[login001: assignment7]$ ls
gauleg.py     int-trap.py        job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679
part1.txt  part2.txt
Homework7.py  James_script.slurm  myjob.e364675     myjob.e364679  myjob.o364678
part1.py       part2.py
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ sbatch James_script.slurm
Submitted batch job 365236
[login001: assignment7]$ squeue -u ayn868
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            365236  compute1 Jamesjob   ayn868  R       0:03      1 c022
[login001: assignment7]$ squeue -u ayn868
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            365236  compute1 Jamesjob   ayn868 CG       0:05      1
[login001: assignment7]$ squeue -u ayn868
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[login001: assignment7]$ ls
gauleg.py     Jamesjob.e365236    job_script.slurm  myjob.e364679  myjob.o364679  part2.py
Homework7.py  Jamesjob.o365236    myjob.e364675     myjob.o364675  part1.py       part2.txt
int-trap.py   James_script.slurm  myjob.e364678     myjob.o364678  part1.txt
[login001: assignment7]$ vim Jamesjob.e365236
[login001: assignment7]$ vim Jamesjob.o365236
[login001: assignment7]$ ls
gauleg.py     Jamesjob.e365236    job_script.slurm  myjob.e364679  myjob.o364679  part2.py
Homework7.py  Jamesjob.o365236    myjob.e364675     myjob.o364675  part1.py       part2.txt
int-trap.py   James_script.slurm  myjob.e364678     myjob.o364678  part1.txt
[login001: assignment7]$ rm Jamesjob.*
Do you really want to run this delete command?
 rm Jamesjob.e365236 Jamesjob.o365236
(y/n)? y
[login001: assignment7]$ ls
gauleg.py     int-trap.py        job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679
part1.txt  part2.txt
Homework7.py  James_script.slurm  myjob.e364675     myjob.e364679  myjob.o364678
part1.py       part2.py
[login001: assignment7]$ vim Homework7.py
```

```
[login001: assignment7]$ vi Homework7.py
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ sbatch James_script.slurm
Submitted batch job 365237
[login001: assignment7]$ ls
gauleg.py     Jamesjob.e365237    job_script.slurm  myjob.e364679  myjob.o364679  part2.py
Homework7.py  Jamesjob.o365237    myjob.e364675     myjob.o364675  part1.py       part2.txt
int-trap.py   James_script.slurm  myjob.e364678     myjob.o364678  part1.txt
[login001: assignment7]$ vim Jamesjob.e365237


Press ENTER or type command to continue
[login001: assignment7]$ vim Jamesjob.o365237
[login001: assignment7]$ rm Jamesjob.*
Do you really want to run this delete command?
 rm Jamesjob.e365237 Jamesjob.o365237
(y/n)? y
[login001: assignment7]$ ls
gauleg.py     int-trap.py        job_script.slurm  myjob.e364678  myjob.o364675  myjob.o364679
part1.txt  part2.txt
Homework7.py  James_script.slurm  myjob.e364675     myjob.e364679  myjob.o364678
part1.py       part2.py
[login001: assignment7]$ rm myjob.*
Do you really want to run this delete command?
 rm myjob.e364675 myjob.e364678 myjob.e364679 myjob.o364675 myjob.o364678
myjob.o364679
(y/n)? y
[login001: assignment7]$ ls
gauleg.py  Homework7.py  int-trap.py  James_script.slurm  job_script.slurm  part1.py  part1.txt
part2.py  part2.txt
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ sbatch James_script.slurm
Submitted batch job 365239
[login001: assignment7]$ squeue -u ayn868
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
[login001: assignment7]$
[login001: assignment7]$ ls
gauleg.py     int-trap.py       Jamesjob.o365239  job_script.slurm  part1.txt  part2.txt
Homework7.py  Jamesjob.e365239  James_script.slurm  part1.py          part2.py
[login001: assignment7]$ vim Jamesjob.e365239
[login001: assignment7]$ vim Jamesjob.o365239
[login001: assignment7]$ ls
gauleg.py     int-trap.py       Jamesjob.o365239  job_script.slurm  part1.txt  part2.txt
Homework7.py  Jamesjob.e365239  James_script.slurm  part1.py          part2.py
```

```
[login001: assignment7]$ vim Homework7.py
[login001: assignment7]$ vi Homework7.py

    x2 = 1

    # Weights(w) and Integration Points(x)
    w=[]
    x=[]
    for i in N:
        x_temp,w_temp  = np.polynomial.legendre.leggauss(i)
        w.append(w_temp)
        x.append(x_temp)

    ### Compute the Quadrature ###

    # Send Data to Workers
    comm.bcast(N, root=0)
    comm.bcast(x, root=0)
    comm.bcast(w, root=0)

    # Collect the results from each worker
    int_results=[]
    for i in range(1,size):
         temp=comm.recv(source=i)
         int_results.append(temp)

    # Compute the table

        # Sort Reults
    SortedResults = sorted(int_results, key=lambda x: x[0])
    print(SortedResults)
```

ChatGPT

ChatGPT

It seems like you're trying to submit jobs using Slurm on a remote machine, specifically the Arc cluster at UTSA. However, you seem to encounter issues with SSH host key verification. This typically happens when the remote host's key changes, which can occur due to various reasons such as system updates or changes in the server configuration.

To resolve this issue, you can do the following:

Update known_hosts file: Remove the offending key from your `known_hosts` file. You can do this manually or use the `ssh-keygen` command with the `-R` option followed by the hostname to remove the entry.

bash

Copy code

```
ssh-keygen -R arc.utsa.edu
```

Replace `arc.utsa.edu` with the hostname of the remote server.
Retry SSH connection: After removing the offending key, try to SSH into the remote machine again. You may be prompted to confirm the authenticity of the host, and upon confirmation, the new host key will be added to your `known_hosts` file.
Ensure correct hostname: Double-check that you're connecting to the correct hostname or IP address of the remote server.
Contact system administrator: If the issue persists or if you suspect a security concern, it's a good idea to contact the system administrator of the remote server for assistance.

Once you've resolved the SSH host key verification issue, you should be able to submit your Slurm jobs without encountering further SSH errors.

03

You

```
## --- Master ---
if rank == 0:

    # Generate Quadrature Number
    N = list(range(1,21))

    # Integration Bounds
    x1 = -1
    x2 = 1

    # Weights(w) and Integration Points(x)
    w=[]
    x=[]
    for i in N:
        x_temp,w_temp  = np.polynomial.legendre.leggauss(i)
        w.append(w_temp)
        x.append(x_temp)

    ### Compute the Quadrature ###
```

```
# Send Data to Workers
comm.bcast(N, root=0)
comm.bcast(x, root=0)
comm.bcast(w, root=0)

# Collect the results from each worker
int_results=[]
for i in range(1,size):
    temp=comm.recv(source=i)
    int_results.append(temp)

# Compute the table

   # Sort Reults
SortedResults = sorted(int_results, key=lambda x: x[0])
print(SortedResults)
```

Here is the master process

ChatGPT

ChatGPT

The code snippet you provided represents the master process in your MPI program. Here's a breakdown of what it does:

Generate Quadrature Number: Creates a list `N` containing quadrature numbers from 1 to 20.
Integration Bounds: Sets the integration bounds `x1` and `x2`.
Compute Quadrature Points and Weights: Computes the Legendre quadrature points and weights for each quadrature number in `N`.
Send Data to Workers: Broadcasts the quadrature numbers (`N`), quadrature points (`x`), and quadrature weights (`w`) to all worker processes.
Collect Results from Workers: Receives the integration results from each worker process and appends them to the `int_results` list.
Compute the Table:

- Sorts the integration results based on the quadrature number.
- Prints the sorted results.

Overall, this code orchestrates the distribution of data to worker processes, collects the results, and prints them in sorted order based on the quadrature number. Make sure the worker processes are correctly handling the received data and sending back the integration results to

the master process. Additionally, ensure that the worker processes are properly synchronized with the master process to avoid deadlocks or race conditions.

03

You

Given the master and worker processes that I have provided, why is worker returning 0?

ChatGPT

ChatGPT

The issue with the worker process returning 0 might be due to incorrect indexing when accessing the broadcasted data. Let's analyze the worker process code snippet you provided:

python

Copy code

```
else:
 print(f"Process {rank} starts") #Displays worker ID

 # Receive Data
 N_local = comm.bcast(None, root=0)
 x_local = comm.bcast(None, root=0)
 w_local = comm.bcast(None, root=0)

 # Takes 3 arrays and isolates data list specific to this worker
 data_local=[N_local[rank-1],x_local[rank-1],w_local[rank-1]]

 # Takes Data and separates list into 3 arrays for weights, x, and N
 w_worker = data_local[2]
 x_worker = data_local[1]
 N_worker = data_local[0]

 # Performs Integration
 sum = 0.0
 for i in range (0,N_worker):
 sum = sum + w_worker[i]*f(x_worker[i])
 result=sum

 # Send Results to Main
 comm.send((rank, result), dest=0)
```

The issue lies in how you are isolating the data specific to each worker. You are using the worker's rank to index the broadcasted arrays (`N_local`, `x_local`, `w_local`). However, in

Python, the rank of the processes in MPI starts from 0, not 1. So, when you are trying to access `N_local[rank-1]`, it's accessing the 0th element for the worker with rank 1, which corresponds to the first element of the broadcasted arrays.

To fix this issue, you should directly use the received broadcasted arrays without subtracting 1 from the rank. Here's the corrected version of the code:

python

Copy code

```
else:
 print(f"Process {rank} starts") #Displays worker ID

 # Receive Data
 N_local = comm.bcast(None, root=0)
 x_local = comm.bcast(None, root=0)
 w_local = comm.bcast(None, root=0)

 # Takes Data and separates list into 3 arrays for weights, x, and N
 w_worker = w_local
 x_worker = x_local
 N_worker = N_local

 # Performs Integration
 sum = 0.0
 for i in range (0, N_worker):
 sum = sum + w_worker[i] * f(x_worker[i])
 result = sum

 # Send Results to Main
 comm.send((rank, result), dest=0)
```

This way, each worker will correctly use the received broadcasted data without any indexing issues, and it should return the correct results to the master process.

03

You

This will not work

ChatGPT

ChatGPT

I apologize for the oversight. Let's correct the code.

Since you're broadcasting the entire lists `N`, `x`, and `w` to all workers, each worker should receive the entire list, not just a part of it. The indexing `rank - 1` is not appropriate here because it's intended for distributing data among workers, not for accessing the received broadcasted data.

Here's the corrected worker process code:

python

Copy code

```
else:
 print(f"Process {rank} starts") # Displays worker ID

 # Receive Data
 N_local = comm.bcast(None, root=0)
 x_local = comm.bcast(None, root=0)
 w_local = comm.bcast(None, root=0)

 # Performs Integration
 sum = 0.0
 for i in range(0, len(N_local)):
 sum += w_local[i] * f(x_local[i])
 result = sum

 # Send Results to Main
 comm.send((rank, result), dest=0)
```

In this corrected version, each worker receives the entire broadcasted lists `N`, `x`, and `w`, and then performs the integration using those lists. There's no need for indexing or isolating specific data for each worker since all workers receive the same data broadcasted from the master process.

03

You
how do I print a table?

ChatGPT

ChatGPT
To print a table in Python, you can use various libraries such as `tabulate`, `prettytable`, or simply format the output using string formatting. Here's an example using the `tabulate` library:

python

Copy code

```
from tabulate import tabulate

# Assuming SortedResults contains the results in the format (quadrature_no, integration_result)
headers = ["Quadrature no.", "Integration Result", "Percent error", "Run time (s)"]

# Compute percent error and run time (assuming they are available)
table_data = []
for quadrature, result, _, _ in SortedResults:
 percent_error = compute_percent_error(result) # You need to implement compute_percent_error function
 run_time = compute_run_time(quadrature) # You need to implement compute_run_time function
 table_data.append([quadrature, result, percent_error, run_time])

# Print the table
print(tabulate(table_data, headers=headers, floatfmt=".6f")) # Adjust floatfmt as needed
```

In this example, `SortedResults` is assumed to be a list of tuples containing the quadrature number, integration result, percent error, and run time. You would replace the placeholders `compute_percent_error` and `compute_run_time` with your actual functions for computing these values. Adjust the headers and data accordingly based on your specific requirements.

03

You

How do you time something in python

ChatGPT

ChatGPT

In Python, you can time the execution of code using the `time` module or the `timeit` module.

Here's how you can do it with each:

Using the `time` module:

python

Copy code

```
import time

start_time = time.time()

# Code to be timed
# For example:
# result = some_function()

end_time = time.time()

execution_time = end_time - start_time
print("Execution time:", execution_time, "seconds")
```

Using the `timeit` module:

python

Copy code

```
import timeit

# Code to be timed, wrapped in a function
# For example:
# def timed_function():
# return some_function()

execution_time = timeit.timeit("timed_function()", setup="from __main__ import
timed_function", number=1)
print("Execution time:", execution_time, "seconds")
```

Both methods measure the time taken to execute the code between the `start_time` and `end_time` markers. The `timeit` module is more suitable for measuring shorter execution times or for benchmarking purposes, while the `time` module is more general-purpose.

03

You

Traceback (most recent call last):
  File "Homework7.py", line 4, in <module>
    from tabulate import tabulate
ModuleNotFoundError: No module named 'tabulate'

ChatGPT

ChatGPT

It seems that you are trying to use the `tabulate` module, but it is not installed in your Python environment. You need to install the `tabulate` module before using it. You can install it using pip, which is the package installer for Python.

You can install `tabulate` by running the following command in your terminal or command prompt:

Copy code

```
pip install tabulate
```

Once installed, you can import and use the `tabulate` module in your Python script without any errors.

03

You
can I pip install into a slurm file?

ChatGPT

ChatGPT
No, you cannot directly run `pip install` commands within a Slurm script (`.slurm` file). Slurm scripts are meant for submitting and managing job execution on a cluster's compute nodes, and they are typically used to specify job parameters such as resources, runtime, and the command to be executed.

To install packages using `pip`, you typically need access to the command line interface of the system where you want to install the package. This means you would need to run `pip install` commands outside of the Slurm script, either directly on the command line or in a separate script.

If you need certain Python packages installed on the compute nodes where your Slurm jobs will run, you should communicate with your system administrator or cluster manager to have those packages installed system-wide or within a specific Python environment accessible to your jobs. Once the packages are available in the environment, you can use them in your Slurm scripts without needing to install them again within the script.

03

You
what are my other options for making a table?

ChatGPT

ChatGPT
If you're unable to use the `tabulate` module, you have a few other options for creating tables in Python:

1. Using `format` Method: You can use the `str.format()` method to format strings and create a simple table-like structure. While this method requires more manual formatting, it can be effective for basic tabular output.

   python

   Copy code

```
data = [("Column1", "Column2", "Column3"),
 (value1, value2, value3),
 (value4, value5, value6)]

for row in data:
 print("{:<15} {:<15} {:<15}".format(*row))
```

2. Using String Manipulation: You can manually manipulate strings to create a table-like structure. This approach gives you full control over the formatting but can be more cumbersome for larger tables.

   python

   Copy code

```
data = [("Column1", "Column2", "Column3"),
 (value1, value2, value3),
 (value4, value5, value6)]

for row in data:
 print(" | ".join(str(item).ljust(15) for item in row))
```

Using Pandas Library: If you have Pandas installed, you can use it to create tables with more flexibility and customization options.

python

Copy code

```
import pandas as pd

data = [("Column1", "Column2", "Column3"),
 (value1, value2, value3),
 (value4, value5, value6)]

df = pd.DataFrame(data[1:], columns=data[0])
print(df)
```

Each of these methods has its own advantages and limitations, so you can choose the one that best fits your requirements and constraints.

03

You
how an I add a tuple to a list of tuples?

ChatGPT

ChatGPT
You can add a tuple to a list of tuples using the `append()` method or by concatenating lists.

Here's how you can do it:

Using `append()` method:

python

Copy code

```
# Existing list of tuples
list_of_tuples = [(1, 'a'), (2, 'b'), (3, 'c')]

# Tuple to add
new_tuple = (4, 'd')

# Adding the tuple to the list
list_of_tuples.append(new_tuple)
```

```
print(list_of_tuples)
```

Using concatenation:

python

Copy code

```
# Existing list of tuples
list_of_tuples = [(1, 'a'), (2, 'b'), (3, 'c')]

# Tuple to add
new_tuple = (4, 'd')

# Adding the tuple to the list
list_of_tuples += [new_tuple]

print(list_of_tuples)
```

Both methods will produce the same result, adding the tuple `(4, 'd')` to the list of tuples.

03

You
can I append to the beginning?

ChatGPT

ChatGPT
Yes, you can append to the beginning of a list in Python using the `insert()` method. Here's how you can do it with a list of tuples:

python

Copy code

```
# Existing list of tuples
list_of_tuples = [(1, 'a'), (2, 'b'), (3, 'c')]

# Tuple to add
new_tuple = (0, 'z')

# Inserting the tuple at the beginning of the list
list_of_tuples.insert(0, new_tuple)
```

```
print(list_of_tuples)
```

This will insert the tuple `(0, 'z')` at the beginning of the list, shifting the existing elements to the right.

03

You
show line numbersi n ivm

ChatGPT

ChatGPT
In Vim, you can display line numbers by typing `:set number` and pressing Enter. To disable line numbers, you can type `:set nonumber` and press Enter. Alternatively, you can use the shortcut `:set nu` to enable line numbers and `:set nonu` to disable them.

03

You
what is the syntax error in these lines? Traceback (most recent call last):
  File "Homework7.py", line 4, in <module>
    from tabulate import tabulate
ModuleNotFoundError: No module named 'tabulate'

ChatGPT

ChatGPT
The error message indicates that the Python interpreter couldn't find the `tabulate` module, which suggests that it's not installed in your Python environment. To resolve this error, you need to install the `tabulate` module using pip.

You can install `tabulate` using the following command in your terminal or command prompt:

Copy code

```
pip install tabulate
```

After installing `tabulate`, you should be able to import it in your Python script without encountering the `ModuleNotFoundError`.

03

You

what is the syntax error in these lines? ## --- Worker ---
else:
    # print(f"Process {rank} starts") #Displays worker ID
    StartTime = time.time()

ChatGPT

ChatGPT

The syntax error in the provided code snippet is the absence of the `if` statement before the `else` statement. In Python, the `else` statement must be preceded by an `if` statement to form an `if-else` block. Here's the corrected version of the code:

python

Copy code

```
## --- Worker ---
if rank != 0:
 # print(f"Process {rank} starts") #Displays worker ID
 StartTime = time.time()
```

With this change, the `else` block is correctly associated with the preceding `if` statement, and the code should execute without syntax errors.

03

You

What about now? ## --- Master ---
if rank == 0:

    # Generate Quadrature Number
    N = list(range(1,21))

    # Integration Bounds
    x1 = -1
    x2 = 1

    # Weights(w) and Integration Points(x)
    w=[]
    x=[]
    for i in N:
        x_temp,w_temp = np.polynomial.legendre.leggauss(i)
        w.append(w_temp)
        x.append(x_temp)

```
    ### Compute the Quadrature ###

    # Send Data to Workers
    comm.bcast(N, root=0)
    comm.bcast(x, root=0)
    comm.bcast(w, root=0)

    # Collect the results from each worker
    int_results=[]
    for i in range(1,size):
        temp=comm.recv(source=i)
        int_results.append(temp)
    # Compute the table

       # Sort Reults
    SortedResults = sorted(int_results, key=lambda x: x[0])
    # print(SortedResults)

       # Print Table
    TableHeaders = ["Quadrature No." , "Integration Result" , "Percent Error" , "Run Time (sec)"]

    SortedResults.insert(0,TableHeaders)
    print(pd.DataFrame(SortedResults[1:], columns=data[0])

## --- Worker ---
else:
    # print(f"Process {rank} starts") #Displays worker ID
    StartTime = time.time()
```

ChatGPT

ChatGPT

There are a couple of issues in the provided code snippet:

- The `import pandas as pd` statement is missing. Without it, the code will raise a `NameError` when trying to use `pd.DataFrame`.
- There's a syntax error in the `print` statement. The closing parenthesis is missing.

Here's the corrected version of the code:

python

Copy code

```
import pandas as pd # Add this line to import pandas module

## --- Master ---
if rank == 0:

 # Generate Quadrature Number
 N = list(range(1,21))

 # Integration Bounds
 x1 = -1
 x2 = 1

 # Weights(w) and Integration Points(x)
 w=[]
 x=[]
 for i in N:
 x_temp,w_temp = np.polynomial.legendre.leggauss(i)
 w.append(w_temp)
 x.append(x_temp)

 ### Compute the Quadrature ###

 # Send Data to Workers
 comm.bcast(N, root=0)
 comm.bcast(x, root=0)
 comm.bcast(w, root=0)

 # Collect the results from each worker
 int_results=[]
 for i in range(1,size):
 temp=comm.recv(source=i)
 int_results.append(temp)
 # Compute the table

 # Sort Results
 SortedResults = sorted(int_results, key=lambda x: x[0])

 # Print Table
 TableHeaders = ["Quadrature No." , "Integration Result" , "Percent Error" , "Run
Time (sec)"]
 SortedResults.insert(0,TableHeaders)
 df = pd.DataFrame(SortedResults[1:], columns=SortedResults[0]) # Corrected this
line
 print(df) # Corrected this line
```

```
## --- Worker ---
else:
 # print(f"Process {rank} starts") #Displays worker ID
 StartTime = time.time()
```

With these changes, the code should work correctly, printing the table with the sorted results.

Make sure you have the `pandas` library installed (`pip install pandas`) to use `pd.DataFrame`.

03

You
is there anything wrong with the else: line?

ChatGPT

ChatGPT
No, there is nothing inherently wrong with the `else:` line. However, the `else:` statement should be followed by an indented block of code that will be executed if the condition in the `if` statement above it evaluates to `False`. In your provided code snippet, the `else:` statement is followed by a comment, which means there is no actual code executed in the `else` block.

If you intend to execute some code in the `else` block, you should remove the comment `# print(f"Process {rank} starts")` or replace it with the actual code you want to execute.

Otherwise, if there is no code to execute in the `else` block, it's perfectly fine to leave it as is.

03

You
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75

```
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax
  File "Homework7.py", line 75
    else:
      ^
SyntaxError: invalid syntax          import math
from mpi4py import MPI
import numpy as np
import pandas
import warnings
import time

# This code was written by James Smith to perform Gaussian Integration

# Suppress Leggauss warning about dividing by Zero
warnings.filterwarnings("ignore", category=RuntimeWarning)

# Start the MPI Process
comm = MPI.COMM_WORLD

# Determine the Total Number of Tasks
size = comm.Get_size()

# Determine the rank (ID) of this task
rank = comm.Get_rank()

# Function to integrate
def f(x):
    return x*math.exp(x)
```

```
N = list(range(1,21))


## --- Master ---
if rank == 0:

    # Generate Quadrature Number
    N = list(range(1,21))
    # Integration Bounds
    x1 = -1
    x2 = 1

    # Weights(w) and Integration Points(x)
    w=[]
    x=[]
    for i in N:
        x_temp,w_temp  = np.polynomial.legendre.leggauss(i)
        w.append(w_temp)
        x.append(x_temp)

    ### Compute the Quadrature ###

    # Send Data to Workers
    comm.bcast(N, root=0)
    comm.bcast(x, root=0)
    comm.bcast(w, root=0)

    # Collect the results from each worker
    int_results=[]
    for i in range(1,size):
         temp=comm.recv(source=i)
         int_results.append(temp)

    # Compute the table

        # Sort Reults
    SortedResults = sorted(int_results, key=lambda x: x[0])
    # print(SortedResults)

        # Print Table
    TableHeaders = ["Quadrature No." , "Integration Result" , "Percent Error" , "Run Time (sec)"]

    SortedResults.insert(0,TableHeaders)
    print(pd.DataFrame(SortedResults[1:], columns=data[0])
```

```
## --- Worker ---

else:

    # print(f"Process {rank} starts") #Displays worker ID
    StartTime = time.time()

    # Receive Data
    N_local = comm.bcast(None, root=0)
    x_local = comm.bcast(None, root=0)
    w_local = comm.bcast(None, root=0)

    # Takes 3 arrays and isolates data list specific to this worker
    data_local=[N_local[rank-1],x_local[rank-1],w_local[rank-1]]

    # Takes Data and separates list into 3 arrays for weights, x, and N
    w_worker = data_local[2]
    x_worker = data_local[1]
    N_worker = data_local[0]

    # Performs Integration
    sum = 0.0
    for i in range (0,N_worker):
        sum = sum + w_worker[i]*f(x_worker[i])
    result=sum

    # Calculate Work Time
    Work_Time = time.time() - StartTime

    # Calculate % Error
    PercentError =100*(result - 0.735758882342885)/0.735758882342885

    # Send Results to Main
    comm.send((rank, result, PercentError, Work_Time), dest=0)

    # Debugging, Delete if code is working I guess
    #if rank==1:
    #    print(N_worker,x_worker,w_worker)
    #    print(sum)
ChatGPT
```
ChatGPT

The issue seems to be with the missing closing parenthesis in the `print` statement just before the `else:` block. You should add a closing parenthesis `)` at the end of that line to resolve the syntax error.

Here's the corrected version:

python

Copy code

```
print(pd.DataFrame(SortedResults[1:], columns=data[0]))
```

After fixing this, the `if-else` block structure should be syntactically correct.

03

You
fuck this

ChatGPT

ChatGPT
I'm sorry to hear that you're frustrated. If you have any specific questions or need further assistance, please feel free to ask. I'm here to help.

03

You
is there anything else wrong with the code?

ChatGPT

ChatGPT
There are a few syntax errors and missing imports in the code you provided. Here are the issues I noticed:

- In the master process section, you are missing a closing parenthesis ")" for the `print(pd.DataFrame(SortedResults[1:], columns=data[0])` line.
- In the worker process section, there is a missing closing parenthesis ")" for the `comm.send((rank, result, PercentError, Work_Time), dest=0)` line.
- You have a syntax error in the worker process section where the `else:` statement is not properly aligned. It should be aligned with the `if` statement.
- You are missing an import statement for the `pandas` library.
- The comment "Sort Reults" contains a typo. It should be "Sort Results".

After fixing these issues, your code should run without syntax errors. If you encounter any further issues or need assistance with specific parts of the code, please let me know!

03

You
thanks

ChatGPT

ChatGPT
You're welcome! If you have any more questions or need further assistance, feel free to ask.

Good luck with your code!

03

You
how do I save this conversation into a pdf?

ChatGPT

ChatGPT
To save this conversation as a PDF, you can follow these steps:

1. Select All Text: Scroll through the conversation and select all the text. You can do this by clicking and dragging your mouse cursor over the entire conversation or by using keyboard shortcuts like Ctrl+A (Cmd+A on Mac) to select all.
2. Copy the Text: Once all the text is selected, right-click and choose the "Copy" option from the context menu. Alternatively, you can use the keyboard shortcut Ctrl+C (Cmd+C on Mac) to copy the text.
3. Open a Text Editor or Word Processor: Open a text editor like Notepad (Windows), TextEdit (Mac), or any word processor like Microsoft Word or Google Docs.
4. Paste the Text: Click inside the text editor or word processor and paste the copied text into a new document. You can right-click and choose the "Paste" option from the context menu or use the keyboard shortcut Ctrl+V (Cmd+V on Mac) to paste the text.
5. Format the Text (Optional): You can format the text as needed, adjust font sizes, styles, or add headings to organize the conversation.
6. Save the Document: Once you have pasted and formatted the text, you can save the document. In most text editors or word processors, you can go to the "File" menu and choose "Save As" to save the document. Choose the PDF format when saving.
7. Name and Choose Save Location: Give your PDF document a name and choose where you want to save it on your computer.
8. Save the Document: Click the "Save" button to save the document as a PDF.

Following these steps will allow you to save the conversation as a PDF document on your computer. If you encounter any issues or need further assistance, feel free to ask!

ChatGPT can make mistakes. Consider checking important information.

?

ChatGPT

Get citation