

Heterogeneous medium - Finite Difference Tests

```
In [1]: import numpy as np

from jax import jit
from jax import numpy as jnp
from jaxdf import FourierSeries
from matplotlib import pyplot as plt

from jwave import FourierSeries
from jwave.acoustics import simulate_wave_propagation, TimeWavePropagationSett
from jwave.geometry import *
from jwave.geometry import circ_mask
from jwave.utils import show_field, show_positive_field
from jaxdf import FiniteDifferences

domain = Domain((128, 128), (0.1e-3, 0.1e-3))
```

```
In [2]: from jwave.logger import set_logging_level
set_logging_level(1)
```

FD with accuracy=8 and smooth_initial=True

```
In [3]: simulation_settings = TimeWavePropagationSettings(smooth_initial=True)
fd_accuracy=8
```

```
In [4]: density = np.ones(domain.N) * 1000
density[50:90, 65:100] = 2300
density = FiniteDifferences(np.expand_dims(density, -1), domain, accuracy=fd_a
```

```
In [5]: medium = Medium(domain=domain, sound_speed=1500.0, density=density)
time_axis = TimeAxis.from_medium(medium, cfl=0.3)
```

```
In [6]: # Defining the initial pressure
N = domain.N
p0 = circ_mask(N, 3, (64, 30))

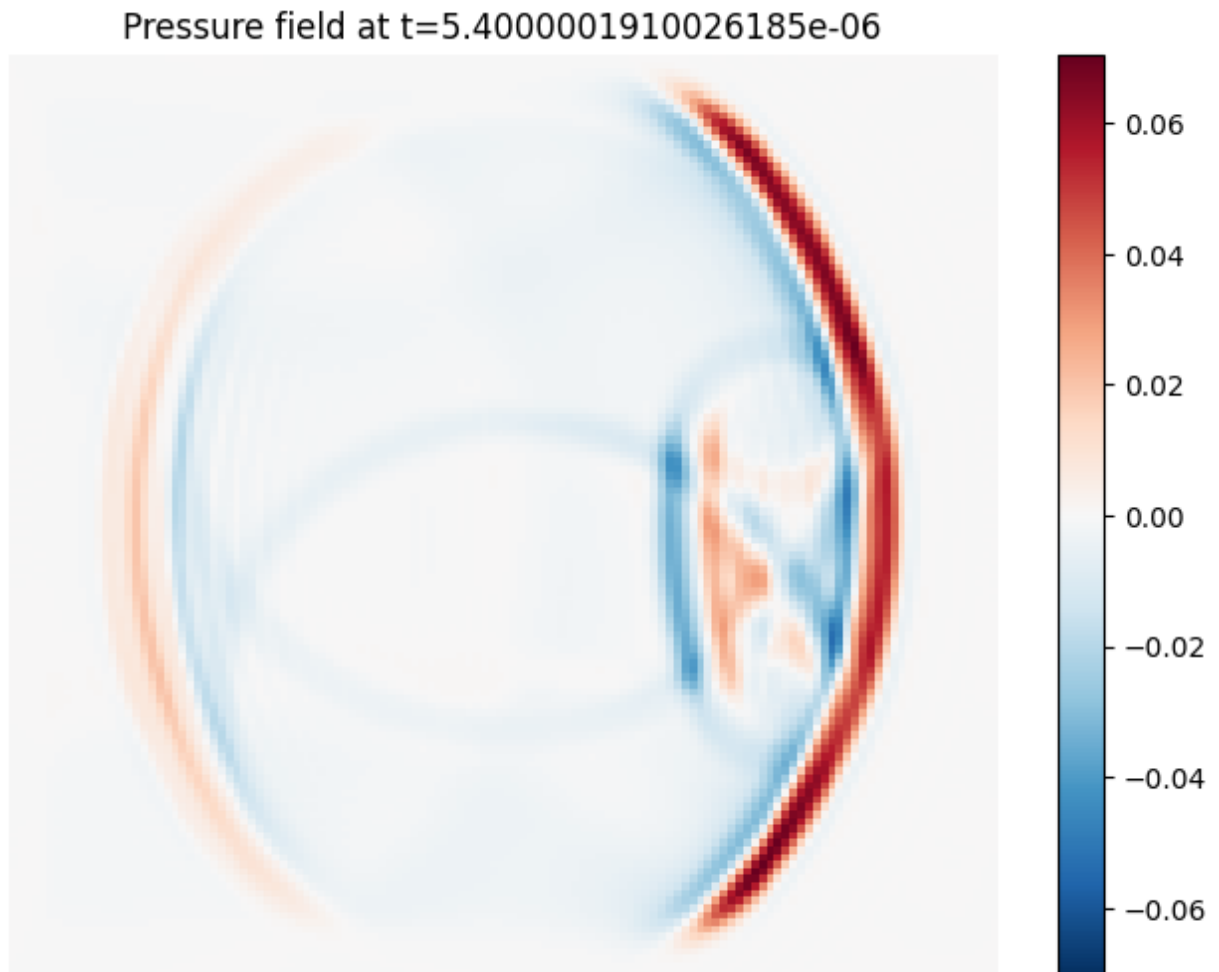
p0 = 1.0 * jnp.expand_dims(p0, -1)
p0 = FiniteDifferences(p0, domain, accuracy=fd_accuracy)
```

```
In [7]: pressure = simulate_wave_propagation(medium, time_axis, settings=simulation_se
```

```
2023-12-14 17:27:24 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:24 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:25 - jwave [DEBUG]: p0 accuracy:8
2023-12-14 17:27:25 - jwave [DEBUG]: pml_rho accuracy:8
2023-12-14 17:27:25 - jwave [DEBUG]: pml_u accuracy:8
2023-12-14 17:27:25 - jwave [DEBUG]: Starting simulation using generic OnGrid
code
2023-12-14 17:27:25 - jwave [DEBUG]: du accuracy:8
2023-12-14 17:27:25 - jaxdf [WARNING]: Deprecation: Currently only the first o
utput of an operator is considered. This will change in a future release. If y
ou need to return multiple outputs, please return a tuple and a None value, fo
r example: ((out1, out2), None). This happened for the operator `mass_conserva
tion_rhs`.
```

```
In [8]: t = 270
show_field(pressure[t])
plt.title(f"Pressure field at t={time_axis.to_array()[t]}")
```

```
Out[8]: Text(0.5, 1.0, 'Pressure field at t=5.4000001910026185e-06')
```



FD with accuracy=8 and smooth_initial=False

```
In [9]: simulation_settings = TimeWavePropagationSettings(smooth_initial=False)
fd_accuracy=8
```

```
In [10]: density = np.ones(domain.N) * 1000
density[50:90, 65:100] = 2300
density = FiniteDifferences(np.expand_dims(density, -1), domain, accuracy=fd_a

N = domain.N
p0 = circ_mask(N, 3, (64, 30))

p0 = 1.0 * jnp.expand_dims(p0, -1)
p0 = FiniteDifferences(p0, domain, accuracy=fd_accuracy)
```

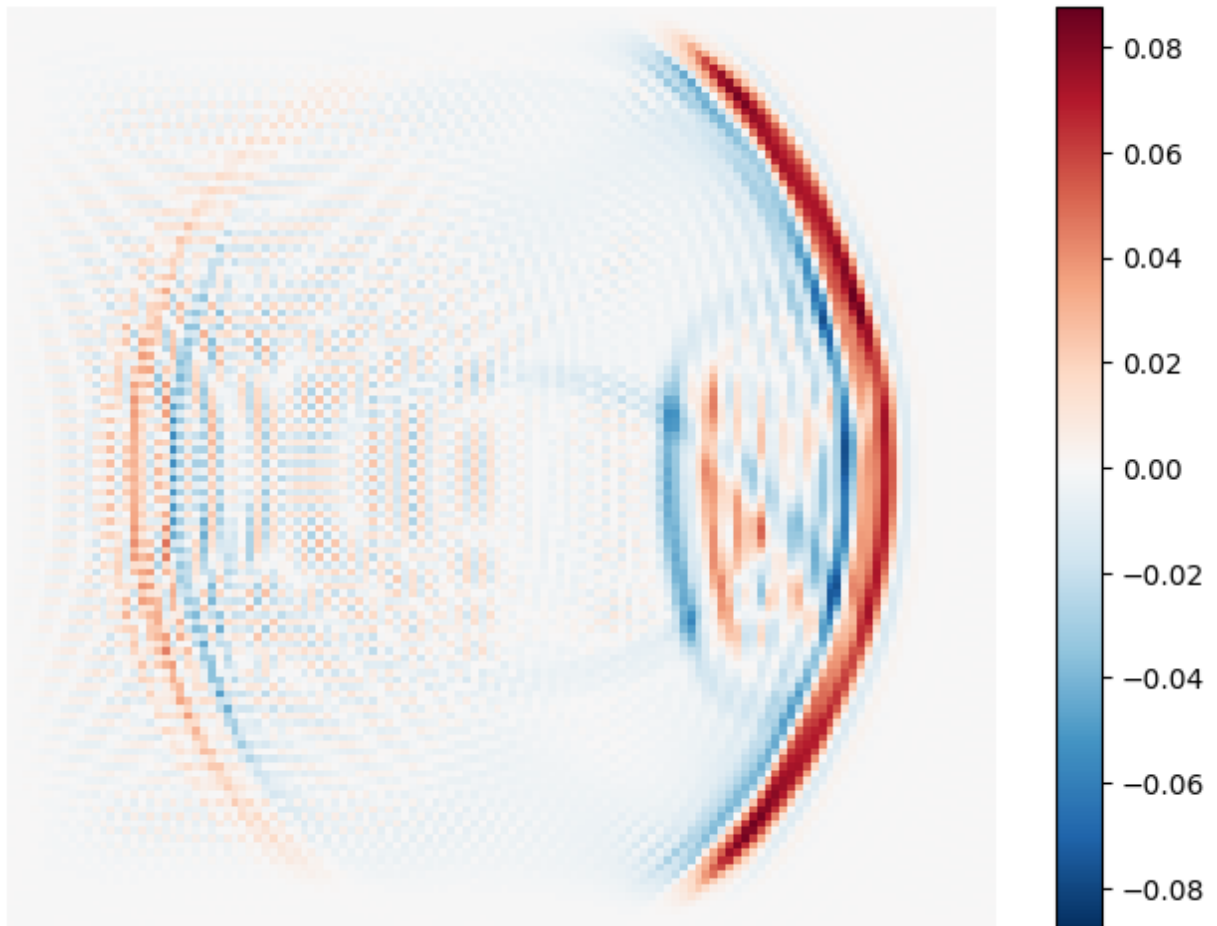
```
In [11]: medium = Medium(domain=domain, sound_speed=1500.0, density=density)
time_axis = TimeAxis.from_medium(medium, cfl=0.3)
```

```
In [12]: pressure = simulate_wave_propagation(medium, time_axis, settings=simulation_se
```

```
2023-12-14 17:27:26 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: p0 accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: pml_rho accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: pml_u accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: Starting simulation using generic OnGrid
code
2023-12-14 17:27:26 - jwave [DEBUG]: du accuracy:8
2023-12-14 17:27:26 - jaxdf [WARNING]: Deprecation: Currently only the first o
utput of an operator is considered. This will change in a future release. If y
ou need to return multiple outputs, please return a tuple and a None value, fo
r example: ((out1, out2), None). This happened for the operator `mass_conserva
tion_rhs`.
```

```
In [13]: t = 270
show_field(pressure[t])
plt.title(f"Pressure field at t={time_axis.to_array()[t]}")
```

```
Out[13]: Text(0.5, 1.0, 'Pressure field at t=5.4000001910026185e-06')
```

Pressure field at $t=5.4000001910026185e-06$ 

FD with accuracy=2 and smooth_initial=False

```
In [14]: simulation_settings = TimeWavePropagationSettings(smooth_initial=False)
         fd_accuracy=2
```

```
In [15]: density = np.ones(domain.N) * 1000
         density[50:90, 65:100] = 2300
         density = FiniteDifferences(np.expand_dims(density, -1), domain, accuracy=fd_a

         N = domain.N
         p0 = circ_mask(N, 3, (64, 30))

         p0 = 1.0 * jnp.expand_dims(p0, -1)
         p0 = FiniteDifferences(p0, domain, accuracy=fd_accuracy)
```

```
In [16]: medium = Medium(domain=domain, sound_speed=1500.0, density=density)
         time_axis = TimeAxis.from_medium(medium, cfl=0.3)
```

```
In [17]: pressure = simulate_wave_propagation(medium, time_axis, settings=simulation_se
```

```
2023-12-14 17:27:26 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: pml accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: p0 accuracy:2
2023-12-14 17:27:26 - jwave [DEBUG]: pml_rho accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: pml_u accuracy:8
2023-12-14 17:27:26 - jwave [DEBUG]: Starting simulation using generic OnGrid
code
2023-12-14 17:27:26 - jwave [DEBUG]: du accuracy:8
2023-12-14 17:27:26 - jaxdf [WARNING]: Deprecation: Currently only the first o
utput of an operator is considered. This will change in a future release. If y
ou need to return multiple outputs, please return a tuple and a None value, fo
r example: ((out1, out2), None). This happened for the operator `mass_conserva
tion_rhs`.
```

```

-----
TypeError                                 Traceback (most recent call last)
Cell In[17], line 1
----> 1 pressure = simulate_wave_propagation(medium, time_axis, settings=simulation_settings, p0=p0)

File ~/Codes/jwave-version-0.2/.venv/lib/python3.11/site-packages/plum/function.py:399, in Function.__call__(self, *args, **kw_args)
    397 def __call__(self, *args, **kw_args):
    398     method, return_type = self._resolve_method_with_cache(args=args)
--> 399     return _convert(method(*args, **kw_args), return_type)

File ~/Codes/jwave-version-0.2/.venv/lib/python3.11/site-packages/jaxdf/core.py:89, in _operator.<locals>.wrapper(*args, **kwargs)
    84 # Log dispatch message
    85 logger.debug(
    86     f"Dispatching {evaluate.__name__} with for types {evaluate.__annotations__}"
    87 )
--> 89 outs = evaluate(*args, **kwargs)
    90 if isinstance(outs, tuple) and len(outs) > 1:
    91     logger.warning(
    92         f"Deprecation: Currently only the first output of an operator is considered. This will change in a future release. If you need to return multiple outputs, please return a tuple and a None value, for example: ((out1, out2), None). This happened for the operator `{evaluate.__name__}`."
    93     )

File ~/Codes/jwave-version-0.2/jwave/acoustics/time_varying.py:495, in simulate_wave_propagation(medium, time_axis, settings, sources, sensors, u0, p0, params)
    492     scan_fun = jax_checkpoint(scan_fun)
    494 logger.debug("Starting simulation using generic OnGrid code")
--> 495 _, ys = scan(scan_fun, fields, output_steps)
    497 return ys

[... skipping hidden 2 frame]

File ~/Codes/jwave-version-0.2/.venv/lib/python3.11/site-packages/jax/_src/lax/control_flow/loops.py:310, in _check_scan_carry_type(body_fun, in_carry, out_carry_tree, out_aval)
    304 else:
    305     differences = '\n'.join(
    306         f' * {component(path)} is a {thing1} but the corresponding component '
    307         f'of the carry output is a {thing2}, so {explanation}\n'
    308         for path, thing1, thing2, explanation
    309         in equality_errors(in_carry, out_carry))
--> 310 raise TypeError(
    311     "Scanned function carry input and carry output must have the same "
    312     "pytree structure, but they differ:\n"
    313     f"{differences}\n"
    314     "Revise the scanned function so that its output is a pair where the "
    315     "first element has the same pytree structure as the first argument."
    316 )
    317 if not all(_map(core.typematch, in_aval, out_aval)):
    318     differences = '\n'.join(

```

```
319     f' * {component(path)} has type {in_aval.str_short()}'
320     ' but the corresponding output carry component has type '
321     f'{out_aval.str_short()}{_aval_mismatch_extra(in_aval, out_ava
l)}\n'
322     for path, in_aval, out_aval in zip(paths, in_aval, out_aval)
323     if not core.typematch(in_aval, out_aval)
```

TypeError: Scanned function carry input and carry output must have the same pytree structure, but they differ:

* the input carry component fields[0] is a <class 'jaxdf.discretization.FiniteDifferences'> with pytree metadata ('params', 'domain'), ('accuracy',), (2,) but the corresponding component of the carry output is a <class 'jaxdf.discretization.FiniteDifferences'> with pytree metadata ('params', 'domain'), ('accuracy',), (8,)), so the pytree node metadata does not match

Revise the scanned function so that its output is a pair where the first element has the same pytree structure as the first argument.