

CLAIMS

1. A method for lowering a computational cost of a computer simulation, comprising:
- 5 performing a checkpointing strategy, wherein the checkpointing strategy comprises:
- storing a correlation checkpoint at a time step, wherein the correlation checkpoint comprises data used to correlate a forward propagated value generated by the computer simulation, and wherein the computer simulation cannot be restarted from the values stored in the correlation
- 10 checkpoint;
- allocating a storage space for the correlation checkpoint; and
- running the computer simulation at each of a plurality of time steps, wherein at each of the plurality of time steps:
- 15 a backwards propagated value from measured data is correlated to the forward propagated value from the computer simulation, wherein the forward propagated value is stored in the correlation checkpoint.
2. The method of claim 1, further comprising:
- storing a full state checkpoint; and
- 20 restarting the simulation from the full state checkpoint.
3. The method of claim 1, further comprising determining an optimum location for storing either a full state checkpoint or a correlation checkpoint, wherein the optimum location is storage space allocated in a graphics processor unit (GPU) memory, a random access memory (RAM), a RAM disk, a disk drive, or any combinations thereof, and wherein
- 25 the determination of the optimum location is based, at least in part, on the access speed of the storage space.
4. The method of claim 1, wherein a storage space used for the correlation
- 30 checkpoint is less than a storage space used for a full state checkpoint.

5. The method of claim 1, further comprising optimizing the checkpointing strategy by steps comprising:

determining a computing cost associated with storing the correlation checkpoint;

5 minimizing the computing cost based, at least in part, on an access speed for each of a plurality of storage units, and the ratio in storage space required for a full state checkpoint to storage space required for the correlation checkpoint; and

generating a table comprising the plurality of time steps and a type of checkpoint to be stored at each of the plurality of time steps.

10

6. The method of claim 5, wherein determining the computing cost comprises:

computing a maximum number of correlation checkpoints that can be stored in a fast storage unit;

15 computing a maximum number of correlation checkpoints that can be stored in a slow storage unit;

if any correlation checkpoints can be stored in fast storage:

compute an optimal number of sweeps if only using fast storage and add to a list;

20 compute a minimum number of sweeps if only using fast storage and add to the list;

if any correlation checkpoints can be stored in slow storage:

compute a minimum number of sweeps if using all types of storage and add to the list;

25 compute an optimal number of sweeps if using all of the fast storage and storing at least one correlation checkpoint in slow storage, add to list if less than last value on list;

add an integer value nearest to each value on the list to the list;

compute the cost for each value on the list; and

return the minimum cost.

7. The method of claim 1, further comprising generating an image of a subsurface region from seismic data.

5 8. The method of claim 1, further comprising history matching reservoir data to a reservoir simulation.

9. A method for comparing collected data to simulated data, comprising:

backwards propagating the collected data across a grid;

10 running a computer simulation to generate forward propagated data across the grid, wherein:

a plurality of correlation buffers are stored during the computer simulation; and

15 the simulation cannot be restarted from any of the plurality of correlation buffers; and

correlating the data stored in the correlation buffers to the backwards propagated data at each point in the grid.

20 10. The method of claim 9, comprising storing a full state checkpoint during the simulation, wherein the simulation can be restarted from the full state checkpoint.

11. The method of claim 9, further comprising:

calculating a cost for storing a full state checkpoint; and

25 determining a time step for storing a full state checkpoint based, at least in part, on the cost.

12. The method of claim 1, wherein the forward propagated data is correlated to backwards propagated data in a next time step.

13. The method of claim 1, further comprising:
storing a plurality of correlation checkpoints in a fast storage medium; and
storing at least one correlation checkpoint in a slow storage medium.

5

14. The method of claim 13, further comprising storing at least one full state
checkpoint, wherein the simulation can be restarted from the full state checkpoint.

15. The method of claim 14, further comprising storing the at least one full state
10 checkpoint in a fast storage medium.

16. A system for correlating simulated data to collected data, comprising:
a processor;
a storage system, comprising data structures representing:

15

measured data;

a propagation algorithm configured to propagate the measured data
backwards-in-time across a grid;

a computer simulation configured to generate forward-in-time simulated data
across the grid; and

20

a memory, wherein the memory comprises code to direct the processor to:

propagate the measured data backwards-in-time across the grid;

populate the grid with forward-in-time simulated data from the computer
simulation;

store a correlation checkpoint at a time step during the computer simulation;

25

and

correlate the backwards propagated data to the simulated data stored in the
correlation checkpoint.

17. The system of claim 16, wherein the processor comprises a single core, a multiple core, a graphics processing unit, or any combinations thereof.

5 18. The system of claim 16, wherein the correlation checkpoint is stored in the memory.

19. The system of claim 16, wherein the memory comprises a random access memory (RAM), a RAM disk, a graphics processing unit memory, or any combinations thereof.

10 20. The system of claim 16, wherein the memory comprises code to store a full state checkpoint at a time step, wherein the computer simulation can be restarted from the full state checkpoint.

15 21. The method of claim 1, wherein propagating simulated excitation pulses forwards in time is performed using a full two-way, wave equation.

20 22. The method of claim 1, wherein propagating simulated excitation pulses forwards in time is performed using based on Kirchhoff, Beam or one-way wave equation migration techniques.

23. A computer implemented method for solving a problem of imaging or inverting seismic data where a time-stepping forward seismic simulation must be accessed in reverse-time order to correlate with a backward time stepping adjoint computation, comprising:

25 determining and implementing a checkpointing strategy that reduces computation time for a fixed amount of available computer memory M , including data storage, by selecting a combination of n_c memory buffers of size C sized to store data from a forward simulation needed to perform a correlation at a checkpointed time step without
30 storing all data needed to restart the simulation from the time step, and n_s memory buffers of

size S sized to store all data needed to restart the forward simulation from a checkpointed time step, wherein $n_c C + n_s S \leq M$, $n_c \geq 1$ and $n_s \geq 0$;

performing the imaging or inverting on the computer using saved data from the selected memory buffers.

5

24. The method of claim 23, wherein the checkpointing strategy is based at least in part on different access speeds of memory resources.

25. The method of claim 23, wherein the checkpointing strategy includes
10 optimizing which time steps are selected as restart checkpoints, for a pre-selected total number of time steps in the forward simulation.

26. The method of claim 25, wherein the optimizing which time steps are selected as restart checkpoints is based on minimizing a computation cost function.

15

27. The method of claim 23, wherein C depends upon what type of correlation is used in the imaging or inverting.

CLAIMS

1. A method for lowering a computational cost of a computer simulation, comprising:

performing a checkpointing strategy, wherein the checkpointing strategy comprises:

selecting a time step for storing a correlation checkpoint, wherein the correlation checkpoint comprises data used to correlate a forward propagated value generated by the computer simulation, and wherein the computer simulation cannot be restarted from the values stored in the correlation checkpoint;

wherein the selecting a time step for storing a correlation checkpoint balances (i) reduction of computer storage requirements from storing a correlation checkpoint instead of a full state checkpoint, without ability to restart the computer simulation, versus (ii) reduction of computation time from storing a full state checkpoint from which the computer simulation can be restarted;

allocating a storage space for the correlation checkpoint; and

running the computer simulation at each of a plurality of time steps, wherein at each of the plurality of time steps:

a simulated value backwards propagated from measured data is correlated to the forward propagated value from the computer simulation, wherein the forward propagated value is stored in the correlation checkpoint.

2. The method of claim 1, further comprising:

storing a full state checkpoint; and

restarting the simulation from the full state checkpoint.

3. The method of claim 1, further comprising determining an optimum location for storing either a full state checkpoint or a correlation checkpoint, wherein the optimum location is storage space allocated in a graphics processor unit (GPU) memory, a random

access memory (RAM), a RAM disk, a disk drive, or any combinations thereof, and wherein the determination of the optimum location is based, at least in part, on the access speed of the storage space.

4. The method of claim 1, wherein a storage space used for the correlation checkpoint is less than a storage space used for a full state checkpoint.

5. The method of claim 1, further comprising optimizing the checkpointing strategy by steps comprising:

- determining a computing cost associated with storing the correlation checkpoint;
- minimizing the computing cost based, at least in part, on an access speed for each of a plurality of storage units, and the ratio in storage space required for a full state checkpoint to storage space required for the correlation checkpoint; and
- generating a table comprising the plurality of time steps and a type of checkpoint to be stored at each of the plurality of time steps.

6. The method of claim 5, wherein determining the computing cost comprises:

- computing a maximum number of correlation checkpoints that can be stored in a fast storage unit;

- computing a maximum number of correlation checkpoints that can be stored in a slow storage unit;

- if any correlation checkpoints can be stored in fast storage:

- compute an optimal number of sweeps if only using fast storage and add to a list;

- compute a minimum number of sweeps if only using fast storage and add to the list;

- if any correlation checkpoints can be stored in slow storage:

- compute a minimum number of sweeps if using all types of storage and add to the list;

compute an optimal number of sweeps if using all of the fast storage and storing at least one correlation checkpoint in slow storage, add to list if less than last value on list;
add an integer value nearest to each value on the list to the list;
compute the cost for each value on the list; and
return the minimum cost.

7. The method of claim 1, further comprising generating an image of a subsurface region from seismic data.

8. The method of claim 1, further comprising history matching reservoir data to a reservoir simulation.

9. A method for comparing collected data to simulated data, comprising:
backwards propagating the collected data across a grid;
running a computer simulation to generate forward propagated data across the grid, wherein:

a plurality of correlation buffers are stored during the computer simulation; and

the simulation cannot be restarted from any of the plurality of correlation buffers; and

correlating the data stored in the correlation buffers to the backwards propagated data at each point in the grid; wherein

time steps in the computer simulation at which the correlation buffers are stored are selected, at least in part, by balancing (i) reduction of computer storage requirements from storing a correlation buffer instead of a full state buffer, without ability to restart the computer simulation, versus (ii) reduction of computation time from storing a full state buffer from which the computer simulation can be restarted.

10. The method of claim 9, comprising storing a full state checkpoint during the simulation, wherein the simulation can be restarted from the full state checkpoint.
11. The method of claim 9, further comprising:
 - calculating a cost for storing a full state checkpoint; and
 - determining a time step for storing a full state checkpoint based, at least in part, on the cost.
12. The method of claim 1, wherein the forward propagated data are is correlated to backwards propagated data in a next time step.
13. The method of claim 1, further comprising:
 - storing a plurality of correlation checkpoints in a fast storage medium; and
 - storing at least one correlation checkpoint in a slow storage medium.
14. The method of claim 13, further comprising storing at least one full state checkpoint, wherein the simulation can be restarted from the full state checkpoint.
15. The method of claim 14, further comprising storing the at least one full state checkpoint in a fast storage medium.
16. A system for correlating simulated data to collected data, comprising:
 - a processor;
 - a storage system, comprising data structures representing:
 - measured data;
 - a propagation algorithm configured to propagate the measured data backwards-in-time across a grid;
 - a computer simulation configured to generate forward-in-time simulated data across the grid; and
 - a memory, wherein the memory comprises code to direct the processor to:
 - propagate the measured data backwards-in-time across the grid;

populate the grid with forward-in-time simulated data from the computer simulation;

store a correlation checkpoint at a selected time step during the computer simulation, said correlation checkpoint containing information sufficient to perform a correlation but insufficient to restart simulation; wherein the selecting a time step for storing a correlation checkpoint balances (i) reduction of computer storage requirements from storing a correlation checkpoint instead of a full state checkpoint, without ability to restart the computer simulation, versus (ii) reduction of computation time from storing a full state checkpoint from which the computer simulation can be restarted; and

correlate the backwards propagated data to the simulated data stored in the correlation checkpoint.

17. The system of claim 16, wherein the processor comprises a single core, a multiple core, a graphics processing unit, or any combinations thereof.

18. The system of claim 16, wherein the correlation checkpoint is stored in the memory.

19. The system of claim 16, wherein the memory comprises a random access memory (RAM), a RAM disk, a graphics processing unit memory, or any combinations thereof.

20. The system of claim 16, wherein the memory comprises code to store a full state checkpoint at a time step, wherein the computer simulation can be restarted from the full state checkpoint.

21. The method of claim 1, wherein propagating simulated excitation pulses forwards in time is performed using a full two-way, wave equation.

22. The method of claim 1, wherein propagating simulated excitation pulses forwards in time is performed using Kirchhoff, Beam or one-way wave equation migration techniques.

23. A computer implemented method for solving a problem of imaging or inverting seismic data where a time-stepping forward seismic simulation must be accessed in reverse-time order to correlate with a backward time stepping adjoint computation, comprising:

determining and implementing a checkpointing strategy that reduces computation time for a fixed amount of available computer memory M , including data storage, by selecting a combination of n_c memory buffers of size C sized to store data from a forward simulation needed to perform a correlation at a checkpointed time step without storing all data needed to restart the simulation from the time step, and n_s memory buffers of size S sized to store all data needed to restart the forward simulation from a checkpointed time step, wherein $S > C$, $n_c C + n_s S \leq M$, $n_c \geq 1$ and $n_s \geq 0$;

wherein the checkpointed time steps are selected, at least in part, by balancing (i) reduction of computer storage requirements from storing a buffer of size C instead of a buffer of size S , without ability to restart the computer simulation, versus (ii) reduction of computation time from storing a buffer of size S from which the computer simulation can be restarted;

performing the imaging or inverting on the computer using saved data from the selected memory buffers.

24. The method of claim 23, wherein the checkpointing strategy is based at least in part on different access speeds of memory resources.

25. The method of claim 23, wherein the checkpointing strategy includes optimizing which time steps are selected as restart checkpoints, for a pre-selected total number of time steps in the forward simulation.

26. The method of claim 25, wherein the optimizing which time steps are selected as restart checkpoints is based on minimizing a computation cost function.

27. The method of claim 23, wherein C depends upon what type of correlation is used in the imaging or inverting.

28. The method of claim 1, wherein the correlation checkpoint consists of less data than required to restart the computer simulation because of at least one of:

(a) computations for the computer simulation are performed with a higher degree of precision than computations for the correlation;

(b) computations for the computer simulation require boundary condition information not required for the correlation;

(c) not all data components that are simulated are correlated; and

(d) the computer simulation uses a denser computational grid than the computations for the correlation.