



Towards a Parallel Coupled Multi-Scale Model of Magnetic Reconnection

Michal Charemza Tony Arber



May 26, 2007





Contents

1 Magnetic Reconnection

2 Lagrangian Remap and Lare2D

3 Adaptive Mesh Refinement Lagrangian Remap and Larma

4 Parallel AMR Lagrangian Remap

5 Further Work





Contents

1 Magnetic Reconnection

2 Lagrangian Remap and Lare2D

- 3 Adaptive Mesh Refinement Lagrangian Remap and Larma
- 4 Parallel AMR Lagrangian Remap
- 5 Further Work





Contents

- 1 Magnetic Reconnection
- 2 Lagrangian Remap and Lare2D
- 3 Adaptive Mesh Refinement Lagrangian Remap and Larma
- 4 Parallel AMR Lagrangian Remap
- 5 Further Work





Contents

- 1 Magnetic Reconnection
- 2 Lagrangian Remap and Lare2D
- 3 Adaptive Mesh Refinement Lagrangian Remap and Larma
- 4 Parallel AMR Lagrangian Remap
- 5 Further Work





Contents

- 1 Magnetic Reconnection
- 2 Lagrangian Remap and Lare2D
- 3 Adaptive Mesh Refinement Lagrangian Remap and Larma
- 4 Parallel AMR Lagrangian Remap
- 5 Further Work





Magnetic Reconnection



- Dependent on large scale boundary conditions
- Affected by small scale non-fluid effects





Magnetic Reconnection



- Dependent on large scale boundary conditions
- Affected by small scale non-fluid effects





Start of time step

- At time step n, solution known on Eulerian grid
- Solution know from from previous step







After Lagrangian Step

- At time step n + 1, solution known on Lagrangian grid.
- Some numerical time dependent method used.







After Remap Step

- At time step n + 1, solution known on Eulerian grid
- Geometrical method used





Solves resistive and Hall MHD equations







What AMR?

Adaptive Mesh Refinement

- Technique for extending a numerical method for solving equations, using different grid resolution is different areas of the domain
- Higher grid resolution only where, and when, desired
- Higher resolution typically in regions of high change of variables





Advantages of AMR?

Speed: Faster than equivalent non-AMR code

Memory: Less memory used than equivalent non-AMR code.

Michal Charemza Multi-Scale Model of Magnetic Reconnection





- Complex code
- Computational time used to communicate between refinement levels
- Computational time used to navigate data structures
- Can be more difficult to parallelise





Disadvantages of AMR

Complex code

- Computational time used to communicate between refinement levels
- Computational time used to navigate data structures
- Can be more difficult to parallelise





- Complex code
- Computational time used to communicate between refinement levels
- Computational time used to navigate data structures
- Can be more difficult to parallelise





- Complex code
- Computational time used to communicate between refinement levels
- Computational time used to navigate data structures
- Can be more difficult to parallelise





- Complex code
- Computational time used to communicate between refinement levels
- Computational time used to navigate data structures
- Can be more difficult to parallelise





Typical Lare2D computational domain

																																			Т		T
Г																																			Т	Т	T
Г																																					
F																																					
F		H	-	-		-			-			-									-		-				-	-		-	-				+	+	-
F		H	-	-	-	-			-			-				-			-		-	-	-	-			-	-		-	-			-	-	+	+
F		H	-	-	-	-			-			-		-		-			-	H	-	-	-	-		-	-	-	-	-	-			-	+	+	-
F			-	-	-	-			-			-		-		-			-	-	-	-	-	-		-	-	-	-	-	-			-	+	-	-
⊢			-	_	_	_			-			_				-			-	-	_	-	_	-			-	_	_	_	-			_	-	+	_
⊢			_	_	_	_						_				_				-	_	_	_	_			_	_	_	_				_	-	-	_
F			_	_	_	_			_			_				_					_	_	_	_			_	_	_	_	_			_	_	4	_
L																																					
L																																					
Г																																			Т	Т	T
Г																																					T
Г																																			-		-
F			-	_		_															_		_					_							-		-
t		H	-	-		-			-			-				-					-		-	-			-	-		-	-				+	+	-
F		\square	-	-	-	-			-			-				-					-	-	-	-			-	-		-	-			-	+	+	+
F		H	-	-	-	-			-			-		-		-			-	H	-	-	-	-			-	-	-	-	-			-	+	-	-
F			-	_	_	_			-			_		-	-	-			-	-	_	-	_	_		-	-	_	_	_	-	-		_	-	-	-
⊢			-	_	_	_			-			_				-			-		_	_	_	-			-	_	_	_	-			_	-	-	-
⊢			_	_	_	_			_			_				_				-	_	_	_	_			_	_	_	_				_	-	-	_
L			_	_		_			_							_					_		_				_	_		_	_				_		
L																																					
L																																					
Г																																			Т	Т	T
Г																																				Т	T
Г																																					
F																																					
t		H	-	-	-				-							-				H	-	-					-	-			-			-	-	+	-
F		H	-	-	-	-			-							-					-		-				-	-		-	-				-	+	+
t		\vdash	-	-	-	-	H	H	-	H	H	-	H		H	-	H	H		H	-	-	-	-	H	\vdash	-	-	-	-	-	H	H	-	+	+	+
F	\vdash	\vdash	-	-	-	-			-			-		-		-				\vdash	-	-	-	-			-	-		-	-			-	+	+	+
H		\vdash	-	-	-	-			-			-				-				\vdash	-	-	-	-			-	-	-	-	-			-	+	+	-
⊢			-	_	_	_			-			_				-			_	-	_	_	_	_			-	_	_	_	-			_	-	-	_
⊢			_	_	_	_			-			_				-					_	_	_	_			_	_	_	_	-			_	-	-	-
⊢			_	_	_	_														\square	_	_	_				_	_		_				_	_	_	
L			_	_		_															_		_					_									
L																																					
E			1																		1															1	
Г																																					T
Г																																					-





AMR Larma computational domain







AMR Larma computational domain

L																														Т	
L																															
L																								_				_			
L																															
L																															
Е						\pm	H		H	E	+	Ŧ	H	H	H	\pm	H		H		\pm		H	H	Ŧ					Т	
L							H			H		H	H	H	H	÷	H		H		÷	Ш	-	+							
Г						Ŧ	H	F		F	Ŧ	F	FF	Ħ	Ħ	Ŧ	H	F	H	F	Ŧ	H	H	Ŧ	H				Т	Т	
E												H	H	H	H						+										
Г	Г					Ŧ	H	F		Æ	Ŧ	Æ	FF	H	H	Ŧ	H	F	Ħ	F	Ŧ	H	H	Ŧ	H				Т	Т	
Г						÷	H	H		Ŧ	÷	Ŧ	H	H	H	÷	H	H	H	H	÷	H	H	H	Ŧ					Т	
E	1					HE	Ю			Æ	H	Æ	HĒ	ΗĒ	HĒ	H		Æ	Ю	Æ	H			£	HE						
Г	Γ.	17	1			F	ΗĨ			FF	H	F	FF	Ħ	Ħ	H		F	ΗĨ	FF	H			Ŧ	ΗĒ		L.,		T	Т	1
Е						Ŧ	H			Æ	Ŧ	Ŧ	H	H	H	Ŧ		H	H	H	Ŧ				H					Т	
Г	Г					Ŧ	Ħ	F		F	Ŧ	Ŧ	F	Ħ	Ħ	Ŧ	Ŧ	F	Ħ	F	Ŧ	H	H	Π	Ŧ					Т	
Г																														Т	
Г	Г													Г	Г														Т	Т	
Е																													Т	Т	
F																														T	
Г																														Т	
г																														+	
Г																														T	
Е																													Т	Т	
F																														T	
Г																													Т	Т	
г																														+	
Г																														Т	
Г																														1	
Г	Γ													Γ	Γ															T	
E																															
г																														T	
Г	Г														Г															Т	
Г																															
Г																														T	
F																														-	
Г																														T	
Г																													-	+	
F																														1	
Г																													+	+	
г																												-	-	+	





AMR Larma computational domain

L																															Т	
L																																
L																									_				_			
L																																
L																																
Е							\pm	H		H	E	+	Ŧ	H	H	H	\pm	H		H		+	H	H	H	Ŧ					Т	
L							+	H			H		H	H	H	H	÷	H		H			H	-	+							
Г		Г					Ŧ	H	F		F	Ŧ	F	H.	-	H		-		H	F	Ŧ	H	H	Ŧ	H				Т	Т	T
													H	H																		
Г	Г	Г					Ŧ	H	F		Æ	Ŧ	Æ	H					F	Ħ	F	Ŧ	H	H	Ŧ	H				Т	Т	T
Г		Γ					÷	H	H		Ŧ	÷	Ŧ	H						H	H	÷	H	H	H	Ŧ					Т	
Е							Ŧ	H	H		Æ	\pm	Ŧ	H					E	H	H	\pm				H					Т	
Г	Γ.	Τ.	Γ.	1			F	ΗĨ			FF	H		FF	F	Ħ	H	Ħ	-	ΗĨ	FF	H	ΗĤ	Ħ	Ŧ	ΗĒ		L.,		T	Т	-
Г								H			H	H		Ħ		H	H			Ħ		H	H	H	Ŧ	H					T	
Г	Г						Ŧ	Ħ	F		F	Ŧ	Ŧ	F	Ħ	F	Ŧ	Ŧ	F	Ħ	F	Ŧ	H	H	Π	Ŧ					Т	-
Г																															Т	
Г	Г	Г	Г												Г															Т	Т	T
Е		Г																												Т	Т	T
F																															T	-
Г		Г																													Т	
Г																															+	
Г																															T	
Е		Г																												Т	Т	T
F																															-	
Г		Г																												Т	Т	
Г																															+	
Г																															Т	-
Г																															1	T
Г	Γ	Γ	Γ												Γ																T	T
Г																																
г																															T	
Г																															1	-
Г																																
Г																															+	-
Г		Г																													-	T
Г																															+	-
Г		Г																												-	+	-
Г		1																												-	+	-
F		t													\square															+	+	-
F		+				-	-						-													-			-	-	+	+





Orszag-Tang Vortex

• Initial Conditions $\rho = 25/9$, p = 5/3

$$v_x = -\sin y \qquad B_x = -\sin y$$

$$v_y = \sin x \qquad B_y = \sin 2x$$

$$v_z = 0 \qquad B_z = 0$$

 $0 \le x, y \le 2\pi$, time from 0 to π , periodic boundary conditions

Simple intial conditions lead to shocks





Test Results



Michal Charemza Multi-Scale Model of Magnetic Reconnection





AMR Patch Placement









Speedup of Orszag-Tang Vortex







Memory usage of Orszag-Tang Vortex







- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





- Choice of architecture (MPI)
- Load balancing
- Communication time
- What and when to communicate
- Processing the results





Domain decomposition with Ghost Patches

L	_		_	_		-	1	-	-	_	_		_	-	_	_		-	_		-	_	_	_	_	_	_	_	_	_	_	_	_	_	-	+	+
⊢	_			_		-	-			_	_		_		_	_		-			-		_		_		_	_		_	_	-	_	_	-	+	-
F	_		-	_		-	-	-	-	_	_	-	_		_	-	-	-	-	-	-		_	-	_	_	_	_	-	_	_	-	_	-	+	+	+
⊢	-	-	-	_		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-	+	+	+
⊢	_	-		_	_	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	_	_	-	_	-	+	+	-
F	-	-	-	-	-	-	-	-	Ŧ	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	Ħ	Ħ	+	-	-	-	-	-	-	+	+	+
⊢	-	-		_		-	-	-	H		H	Ħ	H	Ŧ		H	Ħ	Ħ	Ŧ	Ħ	H	H	H		H	Ŧ	-		-		-	-	-		+	+	-
F	_	-	-	_	_	-	-	-	Ŧ	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	Ŧ					H	Ħ	H	\mp	Ħ	#	+	+	-	_	-	-	_	-	+	+	+
⊢	-	-		_		-	-	-	H.	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	H						Ħ	Ħ	Ŧ	Ħ	Ħ	=	-	-		-	-	-		+	+	-
⊢	_	_	_	_		-	-	-	+	=		Ħ	+	t	=				-			=		+	=	-		-	_	_	_	-	_	_	-	+	_
F	-	-	-	-		-	-	-	H	Ħ	Ħ	Ħ	Ħ	H	Ħ							Ħ	Ħ	Ĥ	Ħ	Ħ	Ħ	Ŧ			-		-		+	+	+
⊢	-		\vdash	-		-	-	-	H	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	H						Ħ	Ħ	H	Ħ	Ħ	Ŧ	Ŧ			-	\square			+	+	-
F	_			_		-	-	-	H.	Ħ	Ħ	Ħ	Ħ	H	Ħ	1	Ħ	Ħ	Ħ	Ħ	Ħ.	Ħ	Ħ	H.	Ħ	Ħ	=	+			_		-		-	+	-
⊢	_			_		-	-	-	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ħ	Ŧ	Ħ	Ħ	=	#	_		_	-	_	_	-	+	-
⊢		-		_		-	-	-	t.	±		Ħ	t	H.	Ť	H	Ħ	Ħ	t t	tt	p±	Η		Ť.	±	Ħ	+	1				-			_	_	_
F	-					-	-	-			-	-	-			-	-	-			-		-				_	_							+	+	+
L				_			-														_						_	_				_	_		_	+	_
F	_			_		-	-		-		_		_			-		-	-		-		_				_	_							-	+	+
⊢									-												-						_	_							_	+	
⊢				_										-					_		_						_								_	_	-
L	_					-	-	-	-		_		_			_		-		-	-		_				_	_							_	+	-
L																					_						_	_							_	+	-
L	_					-	1		-		_		_			_		1			-		_				_	_							_	4	-
L																											_	_							_	+	
L																											_										
L				_			-								_	_											_	_					_	_	_	+	_
L																		1			_						_	_							_	+	-
L	_					-	1	-	-			-	_				-	1			-		_				_	_							_	+	-
L																											_	_							_	+	
L																											_										
L																											_	_							_	+	
L																					_						_	_								4	
L																					_						_	_							_	4	+
L																											_	_								4	
L																											_										
L																											_	_									
L																												_								_	
L																											_	_									
Ľ						1	1	1				1						1		1	1														T	Γ	1



Node 1

Reconnection Lagrangian Remap AMR Parallel AMR Further Work



Domain decomposition with Ghost Patches











Domain decomposition with Ghost Patches









Ghost Patches: Consequences

- Majority of inter-patch communication code can be reused
- Need some way to tell for nodes to tell each other to create (and remove) patches
- Potentially many messages per time step sent to update ghost patches. High MPI latency per message ⇒ lots of time waiting for messages to complete.





AMR Coordinates









AMR Coordinates



Michal Charemza





AMR Coordinates

	3			4	
	1			2	





AMR Coordinates

	3			4		
			 23 -		24	
	-				F -	
	1					
			21		22	2
						\vdash





AMR Coordinates



Michal Charemza





MPI_Struct to combine messages



Michal Charemza





Non Blocking Communication



Michal Charemza





- Implemented: Ghost patches, AMR Coordinates, To do list
- Implemented: MPI Message combining
- Non blocking communication none but patch structure should make this possible
- Load balancing none. Different domain sizes possible. Other methods could require more of a re-write.





- Implemented: Ghost patches, AMR Coordinates, To do list
- Implemented: MPI Message combining
- Non blocking communication none but patch structure should make this possible
- Load balancing none. Different domain sizes possible. Other methods could require more of a re-write.





- Implemented: Ghost patches, AMR Coordinates, To do list
- Implemented: MPI Message combining
- Non blocking communication none but patch structure should make this possible
- Load balancing none. Different domain sizes possible. Other methods could require more of a re-write.





- Implemented: Ghost patches, AMR Coordinates, To do list
- Implemented: MPI Message combining
- Non blocking communication none but patch structure should make this possible
- Load balancing none. Different domain sizes possible. Other methods could require more of a re-write.





Efficiency Metric

Efficiency of run on N processors

$= \frac{\text{Runtime on 1 processor}}{N \times \text{Runtime on } N \text{ processors}}$





Pattern of communication/processing is same on all processes

Time lost due to load imbalance for each computational block

- $= \max_{\text{processors } p} (\text{compute time for } p \text{average compute time})$
- Use timing calls in code

Averages calculated at end of run to minimize communication





- Pattern of communication/processing is same on all processes
- Time lost due to load imbalance for each computational block
 - $= \max_{\text{processors } p} (\text{compute time for } p \text{average compute time})$

- Use timing calls in code
- Averages calculated at end of run to minimize communication





- Pattern of communication/processing is same on all processes
- Time lost due to load imbalance for each computational block
 - $= \max_{\text{processors } p} (\text{compute time for } p \text{average compute time})$
- Use timing calls in code

Averages calculated at end of run to minimize communication





- Pattern of communication/processing is same on all processes
- Time lost due to load imbalance for each computational block
 - $= \max_{\text{processors } p} (\text{compute time for } p \text{average compute time})$
- Use timing calls in code
- Averages calculated at end of run to minimize communication





Efficiency Results

Efficiency of Parallel AMR running Orszag-Tang problem, using 3 levels of refinement on mhdcluster







Communication Time

 For 64 processor run: runtime = 2 hours lost due to load imbalance = 72 min

- Using mpiP profiler
 data sent per node = 82gb
 messages sent per node = 633k
- Using SKaMPI benchmarker on mhdcluster: bandwidth \approx 350mb/s latency \approx 2.45 μ s.
- $lacksymbol{ = } \Rightarrow {\sf Bandwith time} pprox {\sf 4m, latency time} pprox {\sf 1.55s}$
- If perfect scaling runtime would be 17 min: Have 25% of runtime unaccounted for.





- For 64 processor run: runtime = 2 hours lost due to load imbalance = 72 min
- Using mpiP profiler data sent per node = 82gb messages sent per node = 633k
- Using SKaMPI benchmarker on mhdcluster: bandwidth \approx 350mb/s latency \approx 2.45 μ s.
- $lacksymbol{ = } \Rightarrow {\sf Bandwith time} pprox {\sf 4m, latency time} pprox {\sf 1.55s}$
- If perfect scaling runtime would be 17 min: Have 25% of runtime unaccounted for.





- For 64 processor run: runtime = 2 hours lost due to load imbalance = 72 min
- Using mpiP profiler data sent per node = 82gb messages sent per node = 633k
- Using SKaMPI benchmarker on mhdcluster: bandwidth \approx 350mb/s latency \approx 2.45 μ s.
- $\blacksquare \implies$ Bandwith time pprox 4m, latency time pprox 1.55s
- If perfect scaling runtime would be 17 min: Have 25% of runtime unaccounted for.





- For 64 processor run: runtime = 2 hours lost due to load imbalance = 72 min
- Using mpiP profiler data sent per node = 82gb messages sent per node = 633k
- Using SKaMPI benchmarker on mhdcluster: bandwidth \approx 350mb/s latency \approx 2.45 μ s.
- $\blacksquare \implies$ Bandwith time pprox 4m, latency time pprox 1.55s
- If perfect scaling runtime would be 17 min: Have 25% of runtime unaccounted for.





- For 64 processor run: runtime = 2 hours lost due to load imbalance = 72 min
- Using mpiP profiler data sent per node = 82gb messages sent per node = 633k
- Using SKaMPI benchmarker on mhdcluster: bandwidth \approx 350mb/s latency \approx 2.45 μ s.
- $\blacksquare \implies$ Bandwith time pprox 4m, latency time pprox 1.55s
- If perfect scaling runtime would be 17 min: Have 25% of runtime unaccounted for.





Further Work

- Find missing 25%
- Load balance Parallel AMR
- Implemenent non-blocking communication
- Couple to parallel Vlasov