Verification of Iwan-type Plasticity in the Discontinuous-Mesh GPU-powered Wave Propagation Code AWP

Daniel Roten[1], Kim B. Olsen[1], S.M. Day[1] and Y. Cui[2]

[1]San Diego State University, San Diego, CA [2]San Diego Supercomputer Center, La Jolla, CA

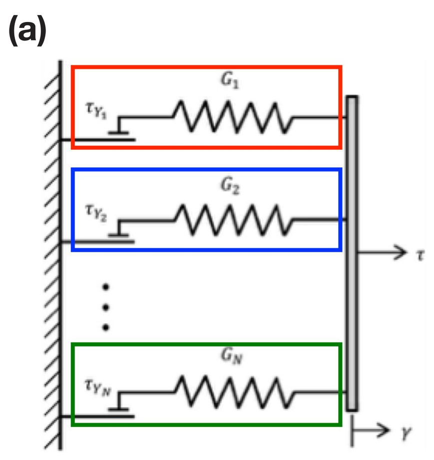
Contact: Daniel Roten (droten@sdsu.edu)

1. Motivation and Scope

- Wave propagation simulations and vertical array observations demonstrate *nonlinear coupling* between source, path and site effects which determines strong ground motions [1].
- Predicting this *shallow crust nonlinearity* is the focus of a technical activity group (TAG) within SCEC.
- One of the TAG's goals is the development, verification and validation of new wave propagation codes which accurately model the shear modulus degradation in soils and which are scalable and efficient enough to resolve a large computational domain.
- We have added support for Iwan-type nonlinearity in the finite difference wave propagation code AWP.
- Iwan nonlinearity was first implemented in the CPU code of AWP and verified against independent codes under previous SCEC awards [2].
- Iwan-type nonlinearity has now been implemented in the GPU-powered, discontinuous mesh (DM) version AWP-GPU-DM.
- Several bugs in AWP-GPU-DM with Iwan plasticity have recently been identified and fixed during the code verification phase.

2. Overlay Concept

- AWP-GPU-DM uses the overlay concept [4] to model Masing unloading and reloading behavior.
- It tracks a series of parallel-series von Mises (J2) yield surfaces (Fig. 1), which in combination reproduce the behavior of a general class of material models originally conceived by Iwan [3].
- Each yield surface is characterized by its own Lamé parameters and requires a separate stress and plasticity update (Fig. 2)



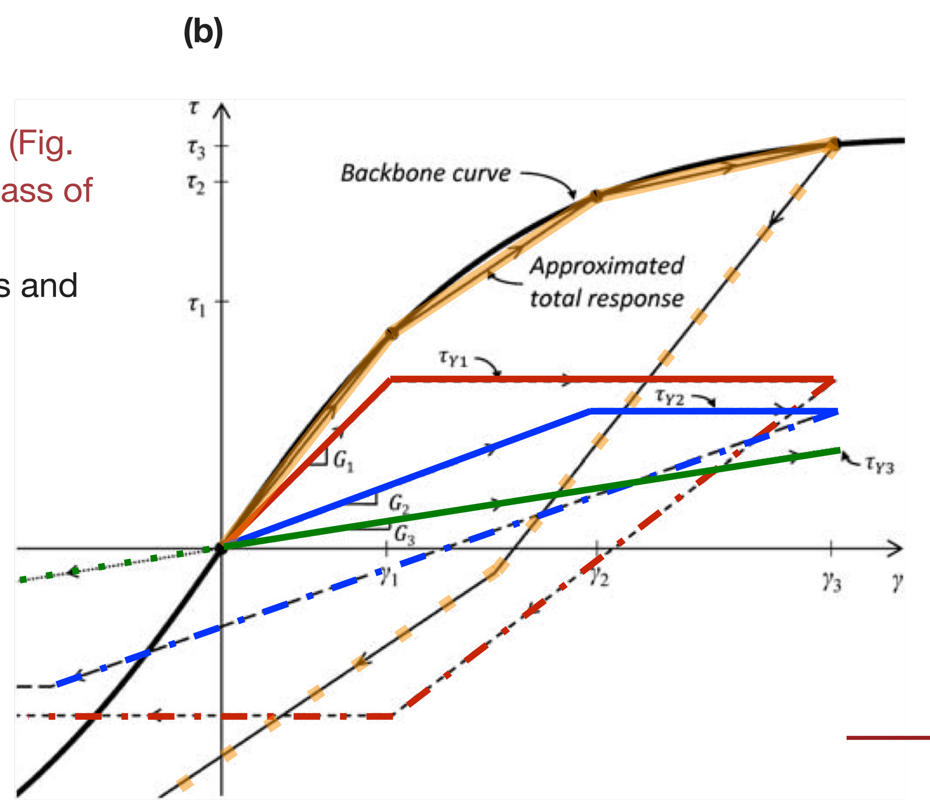


Figure 1. *(a)* Parallel-series configuration of spring-slider arrangement in 1D Iwan model.
(b) Stress-strain behavior of 3 elasto-plastic elements [modified from 4].

3. Implementation Details

- We have developed CUDA kernel functions which perform stress and J2 plasticity updates and compute the overlay velocity field (Fig. 2).
- To conserve GPU memory, material properties pertaining to each surface are re-calculated on demand, rather than permanently stored.

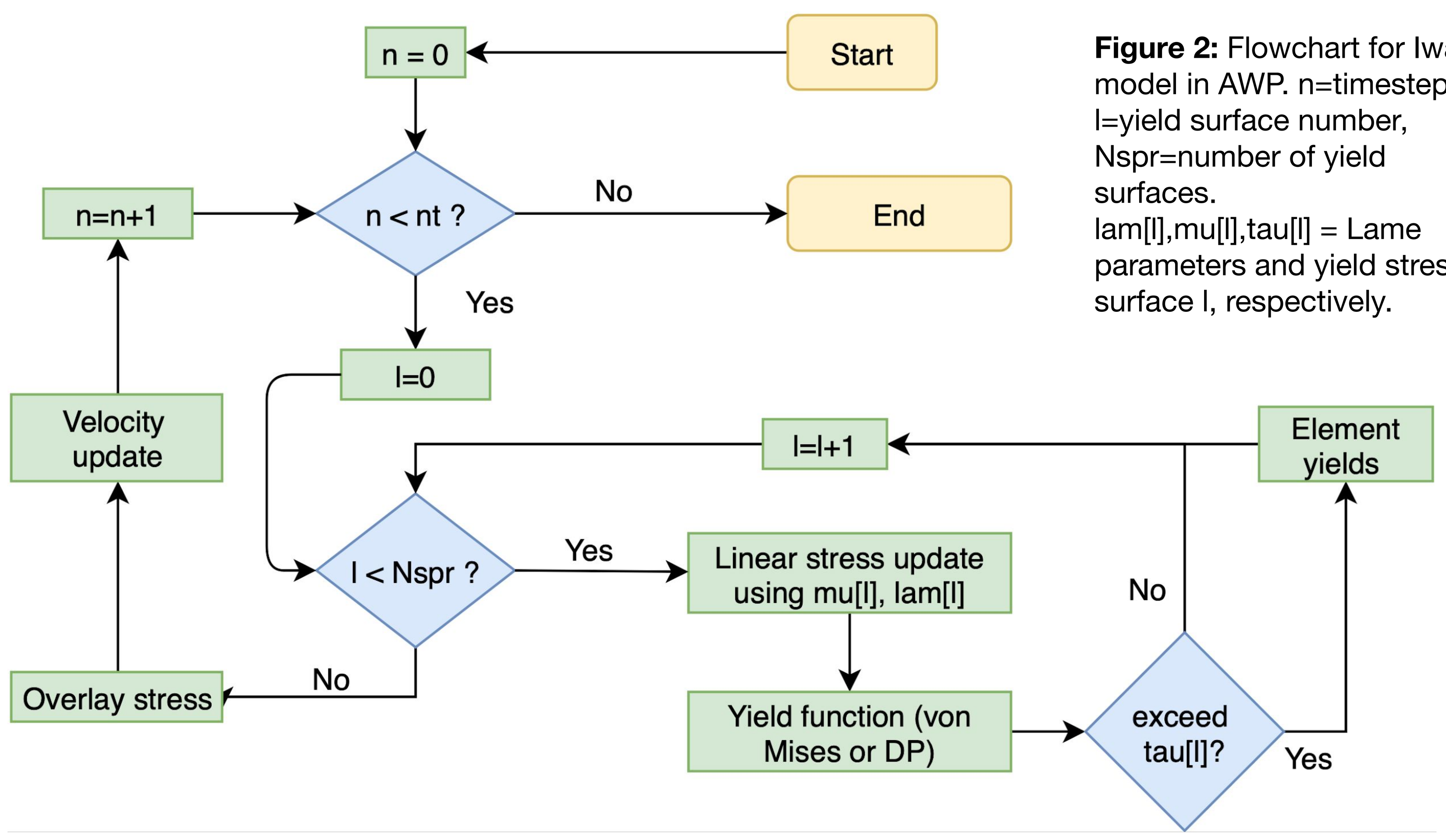


Figure 2: Flowchart for Iwan model in AWP. n=timestep, l=yield surface number, Nspr=number of yield surfaces. lam[l],mu[l],tau[l] = Lame parameters and yield stress of surface l, respectively.

4. Code Verification

- We verified the implementation of Iwan plasticity for the 2D case by carrying out a P-SV simulation and computing reference solutions using Noah2D [5] and the CPU version of AWP [2].
- A vertically incident, transverse (perpendicular to valley axis) plane wave composed of a Ricker wavelet with a central frequency of 2 Hz was used as a source.
- The structure consists of a sloping and sine-shaped valley filled with horizontally layered sediments (Fig. 5b).
- The horizontal discretization was set to 2 m in AWP (CPU and GPU versions) and 1 m in Noah2D.
- Nonlinear time series obtained with the three codes are very similar in both shape and amplitude.
- All codes predict significant reductions in peak ground velocities and duration inside the valley center.

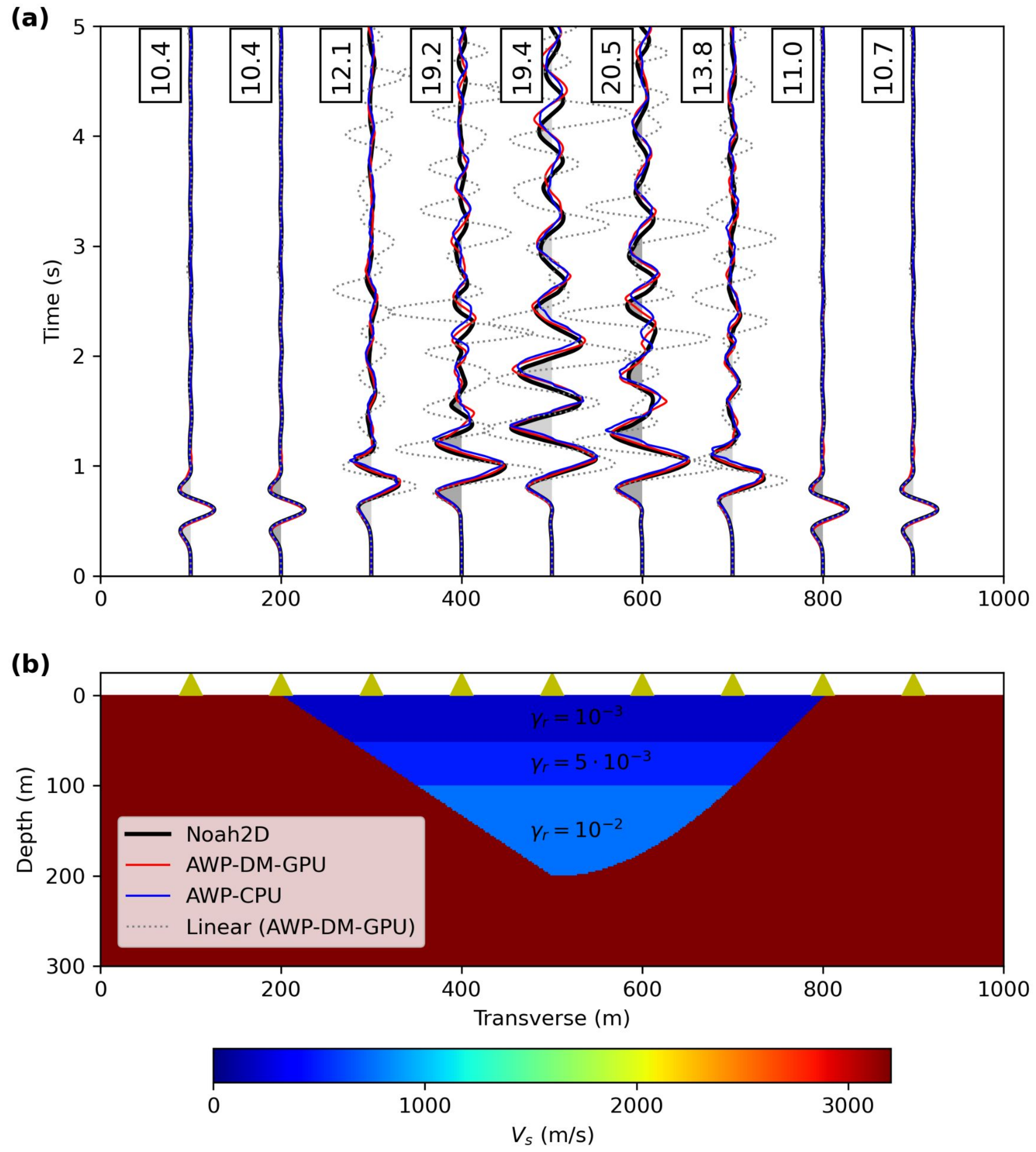


Figure 5. Comparison of transverse velocity time series obtained from 2D nonlinear P-SV propagation inside a sediment-filled valley. (a) Black semi-filled wiggles show the reference solutions obtained with Noah2D; red and blue lines show solutions by the GPU and CPU version of AWP, respectively. Numbers show peak velocities obtained by the reference solution in cm/s (b) Cross-section through valley with shear-velocity (v_s), reference strain (γ_r) and locations of stations shown in (a).

5. Parallel Efficiency

- We have benchmarked the parallel efficiency of AWP-GPU-DM with Iwan nonlinearity on OLCF Summit
- The benchmark was carried out using a DM with 3 different mesh sizes and 10 yield surfaces in the uppermost grid.
- We measured a parallel efficiency of 94.7% on 16,384 Volta V100 GPUs (Fig. 5, Table 1)

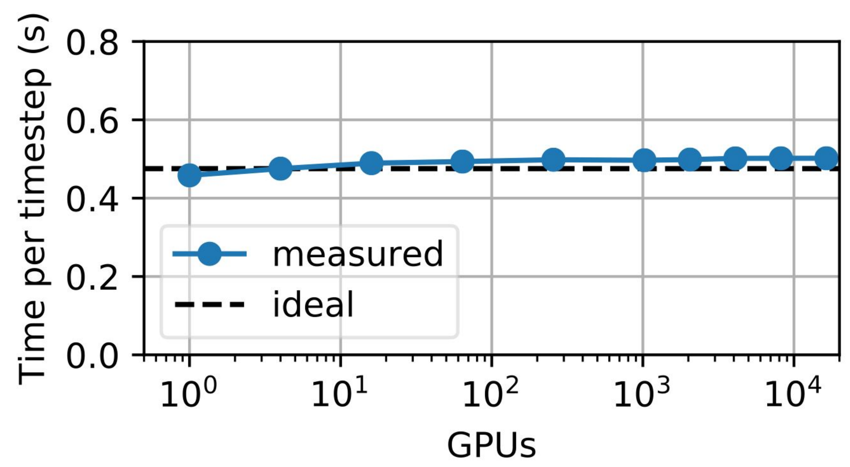


Figure 5: Weak scaling of AWP-GPU-DM with Iwan nonlinearity on OLCF Summit.

GPUs	Time per Timestep	Efficiency (%)
1	0.4583	103.7
4	0.4754	100.0
16	0.4894	97.1
64	0.4935	96.3
256	0.4981	95.4
1,024	0.4968	95.7
2,048	0.4985	95.4
4,096	0.5016	94.8
8,192	0.5018	94.7
16,384	0.5019	94.7

Table 1: Time per timestep and parallel efficiency from AWP-GPU-DM weak scaling test with Iwan nonlinearity on OLCF Summit.

6. Summary and Outlook

- Multi-surface nonlinearity using the Iwan model was implemented In both the CPU and GPU versions of the AWP finite difference code.
- The correctness of the algorithm was verified by comparing solutions of 1D and 2D benchmarks against reference waveforms computed using Noah [2,6].
- CUDA kernels required for the Iwan model in the GPU code were verified by comparison against the CPU version of AWP.
- The high computational density of Iwan plasticity results in very good weak scaling performance on several 1,000 GPUs.

Selected References

1. Day, S., Oral presentation at 2019 SCEC annual meeting (AM)
2. Roten, D.,Olsen, K.B., Day., S. & Cui, Y., Poster during 2018 SCEC AM
3. Iwan, J. Appl. Mech, **34**(4), 1967
4. Kaklamanos, J., Dorfmann L. & Baise, L., SRL, **86**(2A), 2015
5. Gélis & Bonilla, GJI, **191**(3), 2012

SC/EC AN NSF+USGS CENTER

NSF

USGS science for a changing world

SAN DIEGO STATE UNIVERSITY

SDSC SAN DIEGO SUPERCOMPUTER CENTER

Acknowledgments

Weak scaling tests were performed on OLCF Summit under an INCITE (DOE) program allocation award. This project is funded by SCEC award 20166.