

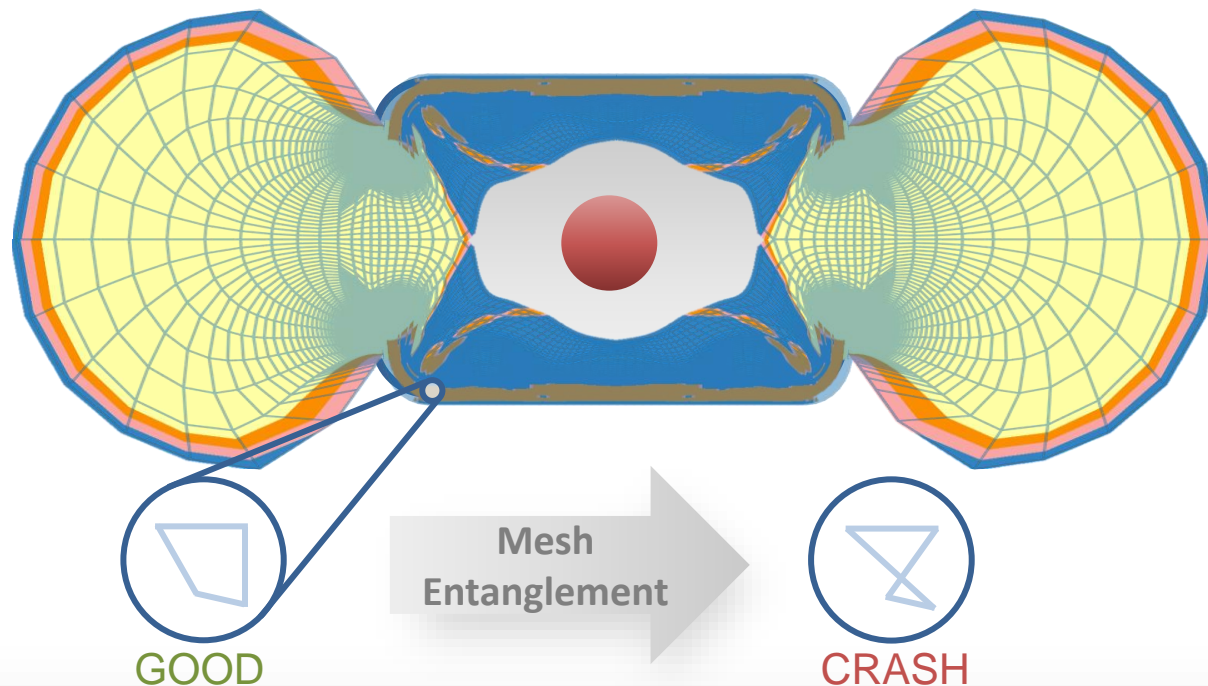


MALT: Machine ALE Learning Technology

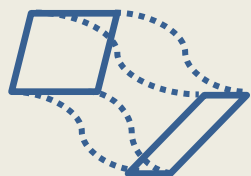
Christopher Yang, Jay Salmonson, Chris Young,
Joe Koning, Luc Peterson
Weapons and Complex Integration/Design Physics



HYDRA¹ Simulates ICF Hohlräume

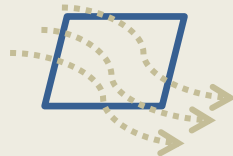


Lagrangian



Mesh moves and deforms

Eulerian



Fluid flows across fixed mesh

Relax



Refine a highly deformed mesh

Super Relax

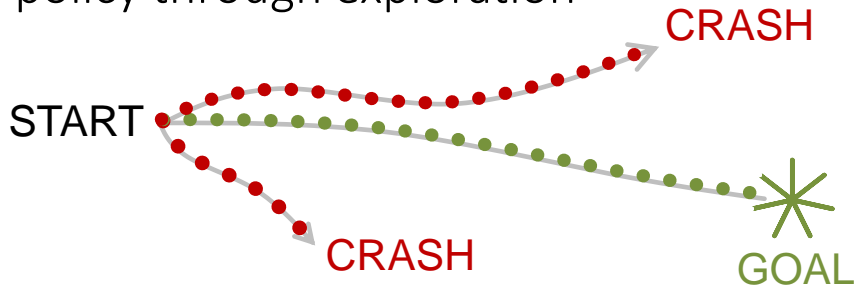
Relax

+

Eulerian

Reinforcement Learning (RL) Roadmap

Finding the best mesh management policy through exploration



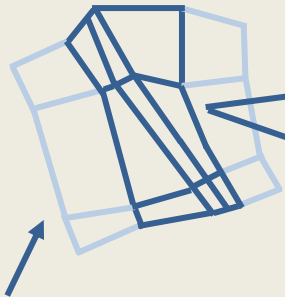
Every time step, the RL agent:

- a. chooses relaxation directives
- b. observes the result of the next step
- c. receives a reward based on mesh quality

Goal: Maximize cumulative reward

1 Search

Identify problems with k-means clustering:

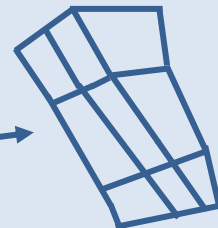


Acceptable zones are not relaxed

✓ Completed

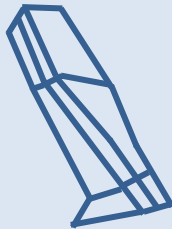
2 Reward

After n time steps, compare:



RL Agent Policy

Issue reward based on mesh improvement



No Action*

*predicted by neural net^{1,3}

✓ Nearing Completion

3 Train

Offline learning:

Merlin + Hydra generates database for off-policy actor-critic agent

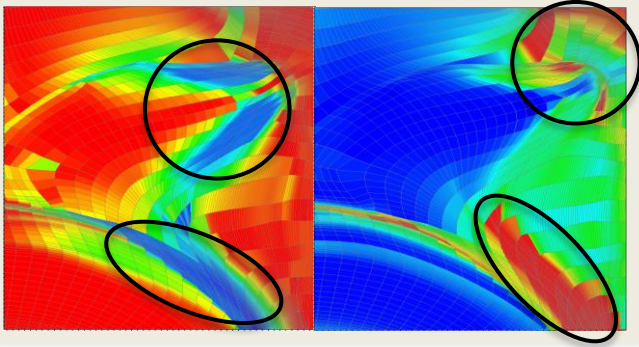
Online learning:

On-policy actor-critic algorithm

□ Future work

1 Metrics Evaluate Mesh Quality³

Scaled Jacobian Condition Number



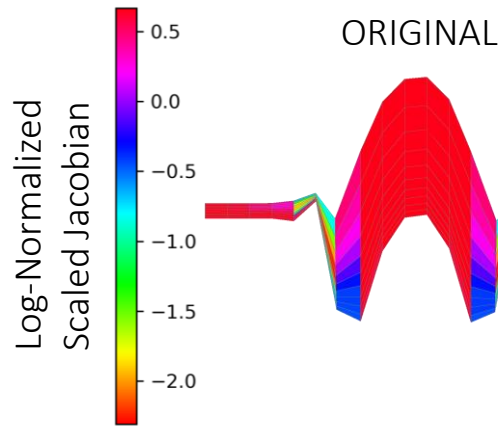
2 Rewards Assess RL Policies

Moment $M < 0$ if, on average, metrics improve

$$\Delta = M_{\text{no action}} - M_{\text{RL}}$$

$$\text{Reward} = \left(\sum_{\text{metrics}} s \Delta \right) - s' (\# \text{relaxed nodes}) \quad (s, s' \text{ are scale factors})$$

3 Example: Scaled Jacobian

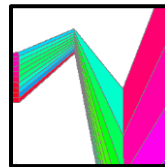


No Action

Manual Policy
(to be replaced by RL)



LAGRANGE
RESULT:



RELAXATION
RESULT:

Lagrangian (Lag.) Moment
+42.6 (neural net)
+41.5 (actual)

$\Delta = \text{Lag.} - \text{Rel. Moment}$
+46.0 (neural net)
+44.9 (actual)

Relaxation (Rel.) Moment
+0.311 (actual)

References & Code

- [1] M. M. Marinak, et al. Phys Plasmas 8, 2275 (2001)
- [2] S. Jégou, et al. PROC CVPR IEEE, 11 (2017)
- [3] J. Salmonson, et al. Bulletin of the American Physical Society, BM10.3 (2018)
- [4] B. Fortuner. *One Hundred Layers Tiramisu*. https://github.com/bfortuner/pytorch_tiramisu (2018)

Combine with other metrics
to calculate rewards!



Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.