

# The Hidden Gem of LAMMPS

Dr. Axel Kohlmeyer

Information and Telecommunication Section  
The Abdus Salam International Centre  
for Theoretical Physics

<http://sites.google.com/site/akohlmey/>

**akohlmey@gmail.com**



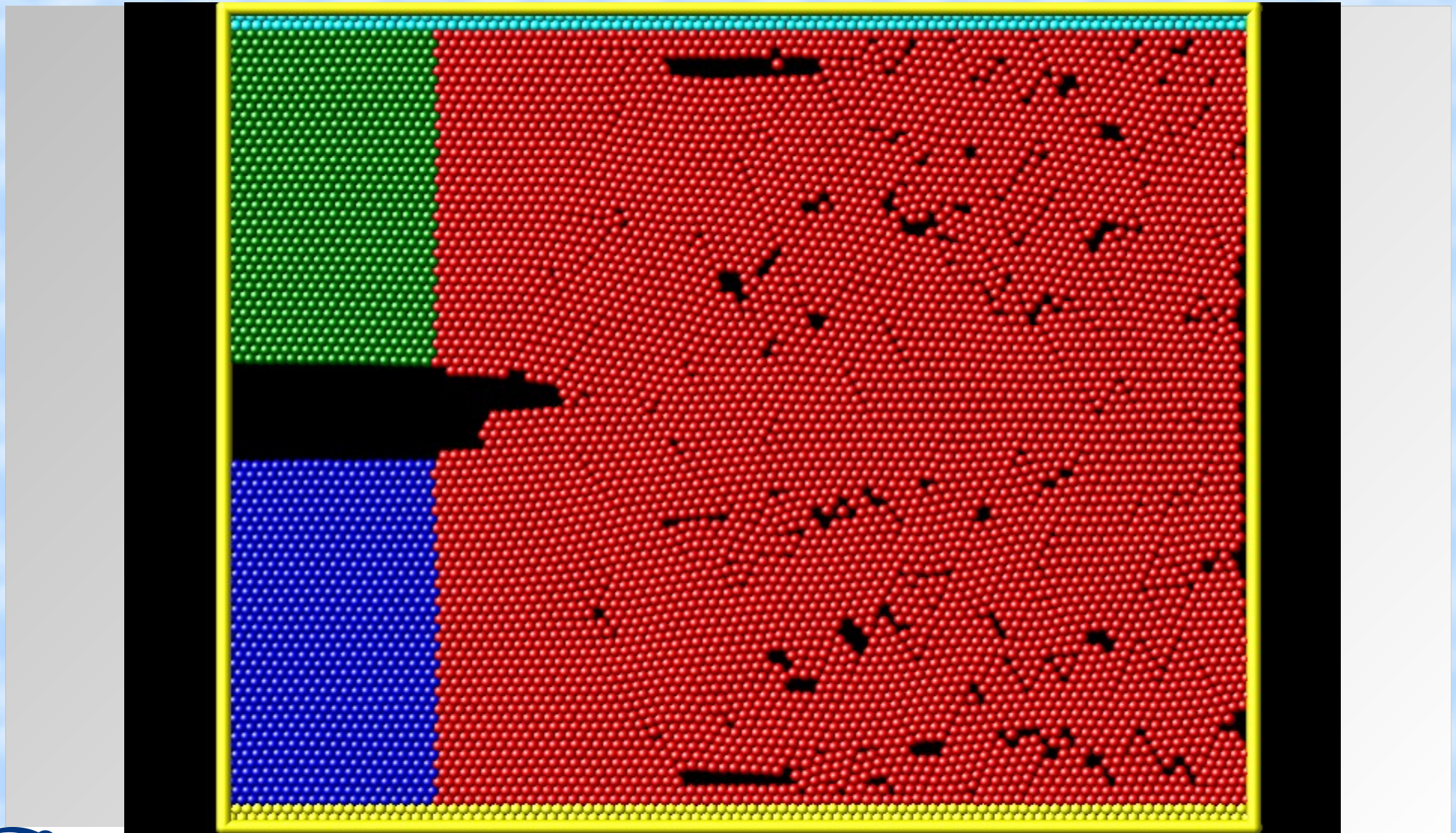
# The `dump image` command

- Included in LAMMPS since 06/2011  
Author: Nathan Fabian (Sandia)
- Outputs 24-bit RGB images to “netpbm” format
- JPEG format available as compile time option
- Format selection based on filename extension:  
  . `jpeg` or . `jpg` -> JPEG, any other -> PNM
- Dump filename format: `image.*.pnm`  
=> one image per snapshot  
=> '\*' replaced by current timestep number

# Advantages & Disadvantages

- Advantages:
  - No external visualizer (useful for quick checks)
  - Parallel rendering (with room for improvement)
  - No large data transfers; no (slow) remote display
  - Compressed output with JPEG images
  - Access to per atom properties, computes
- Disadvantages:
  - Cannot change anything after the fact
  - No periodic display, no complex visualizations

# Simple Example: Crack



# Settings used

Camera zoom

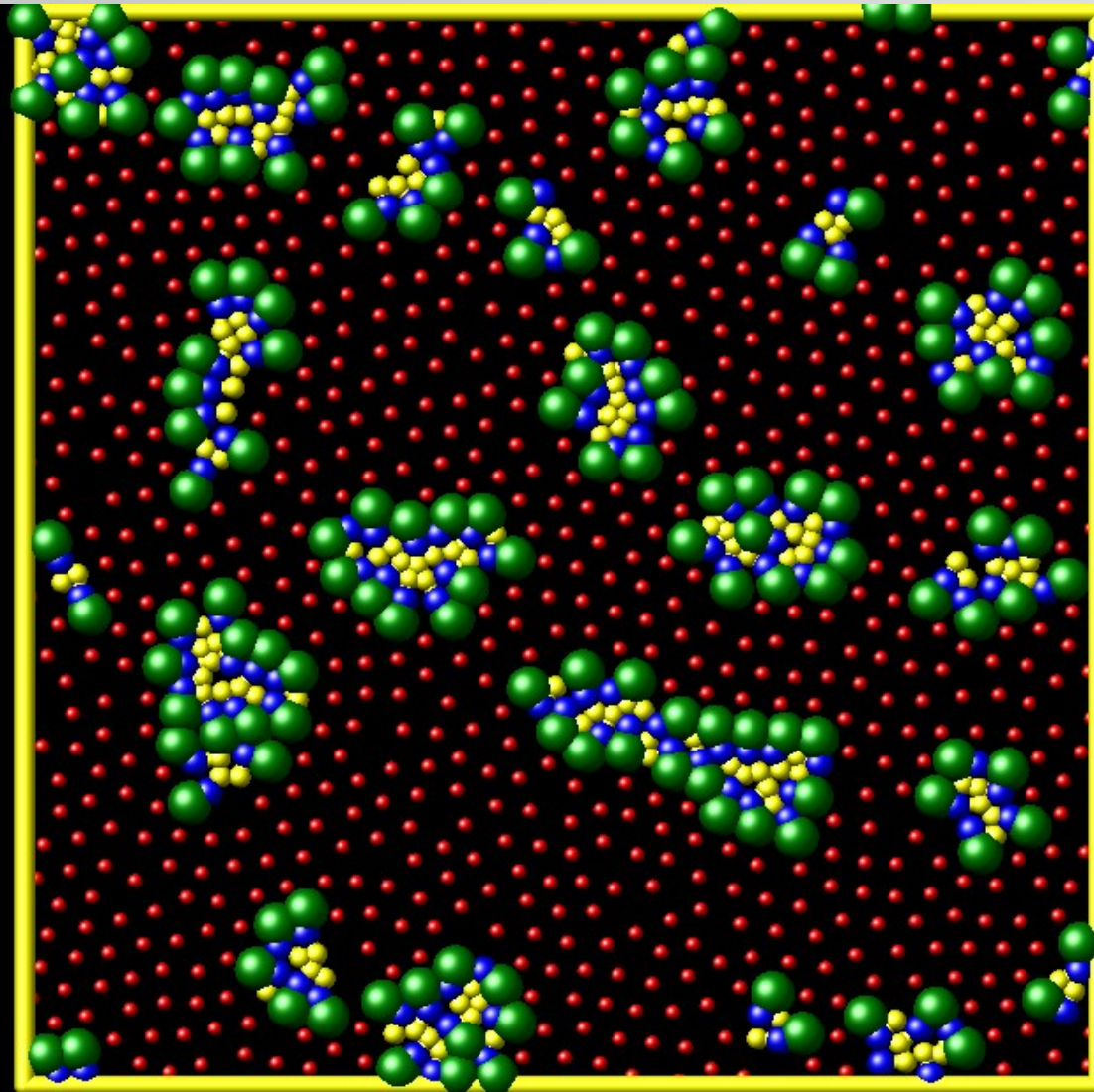
Color by type

```
dump 1 all image 1500 image.*.pnm type type &  
zoom 1.8 adiam 1.4 size 1024 768
```

Fixed diameter for all

Image size

# Simple Example: Micelle



# Settings used

```
dump      1 all image 10000 image.*.pnm type type &  
          zoom 1.6 size 1024 768  
dump_modify 1      adiam 1 0.5 adiam 2 1.5 &  
          adiam 3 1.0 adiam 4 0.75
```

Diameter set individually  
for each type

# Making Movies

- Convert stream of images into animation
- Call converter tools from LAMMPS via **shell**
  - Simplest option: **convert** tool of ImageMagick  
=> animated GIF, MPEG1 => 8-bit color, low quality
  - HD quality possible via FFmpeg or Mencoder using DivX/MPEG-4 or H.264 compression:  

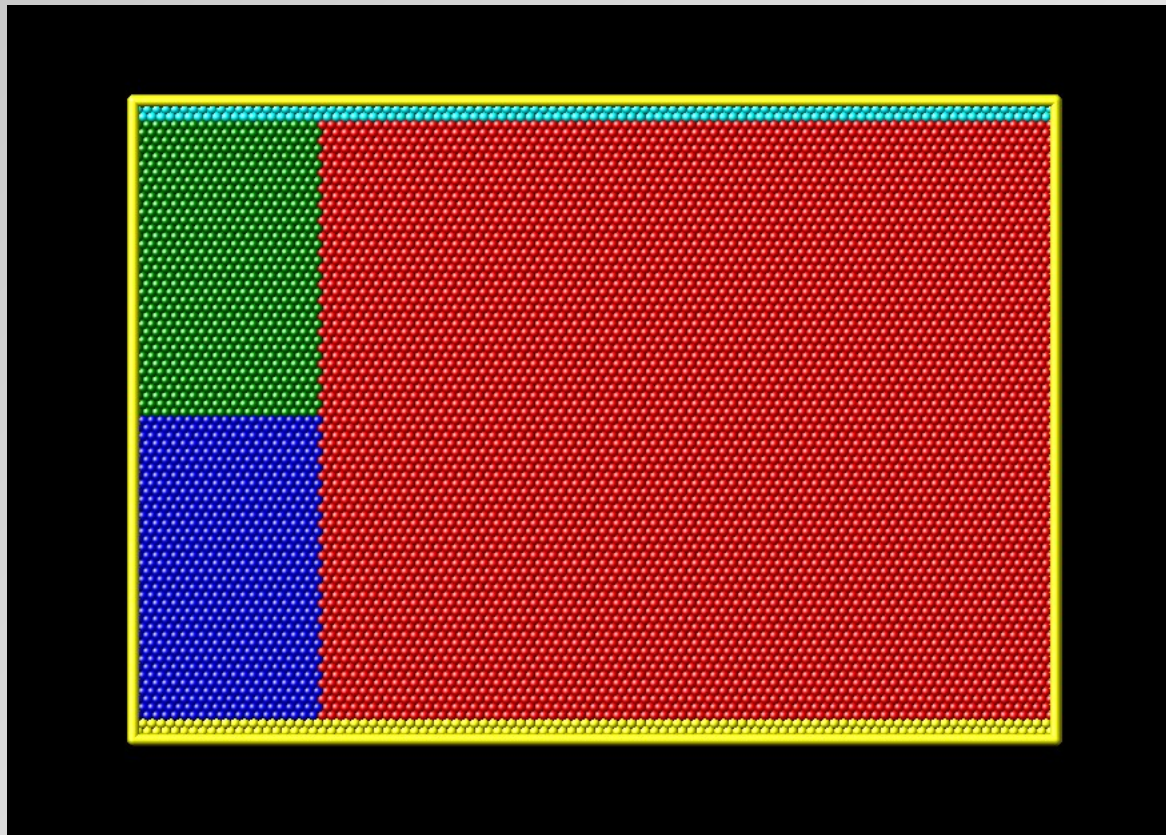
```
shell ffmpeg -y -an -i:v snap-movie.%d.ppm &  
-r 24 -b:v 2400k -c:v libx264 crack.mp4
```
  - FFmpeg requires consecutive image numbering:  

```
shell sh -c 't=0 ; for s in image.*.ppm; do  
mv $s image.$t.ppm ; t=`expr $t + 1` ; done'
```



# Example Movie: Crack

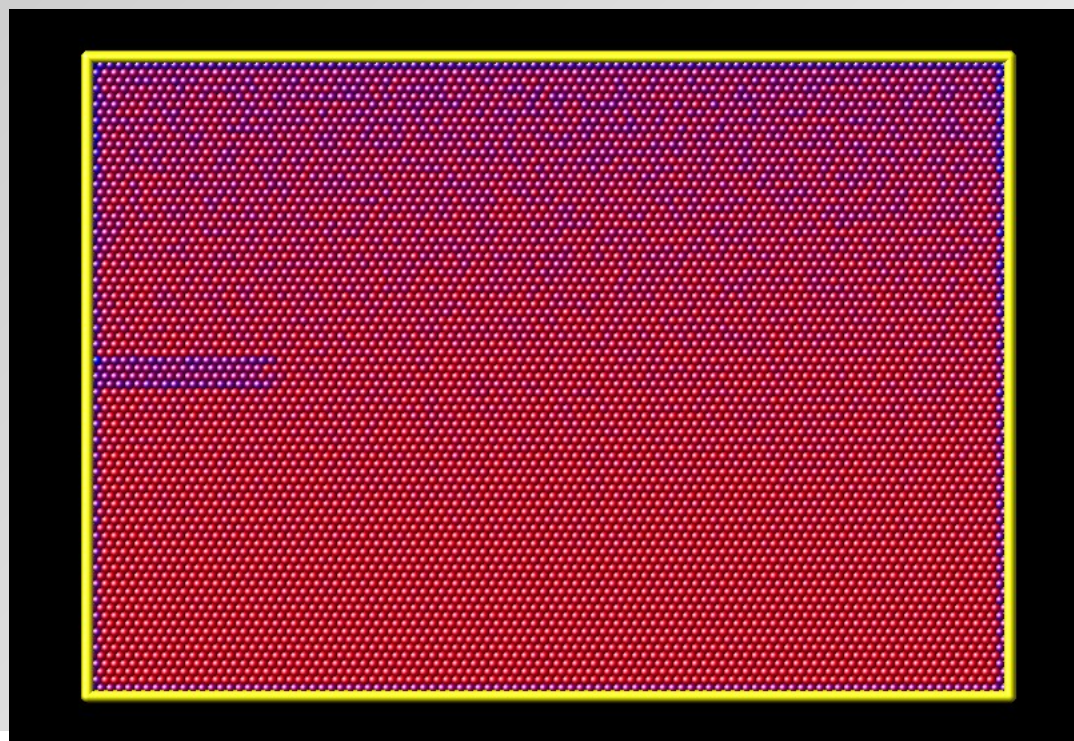
- 960x640px, 24 fps, H.264@2400 kbps, 3.7MB



# Example Movie: Color by Compute

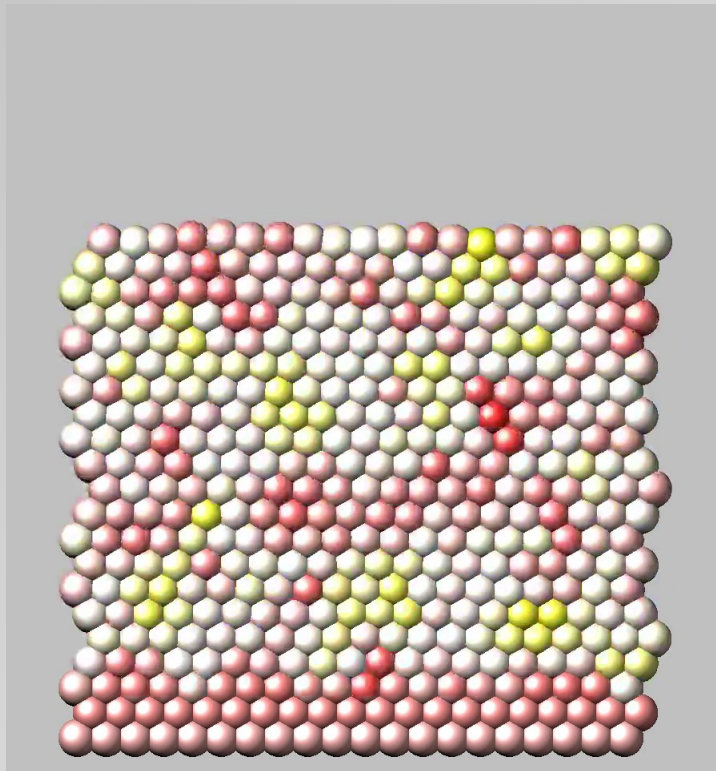
- Using colorscale for dynamic atomic property:  

```
compute s all stress/atom  
variable spa atom c_s[1]+c_s[2]+c_s[3]  
dump 1 all image 100 image.*.pnm v_spa type ...
```



# More Example Movies

- Indent example with fixed color scale



- Sputter example with ambient occlusion

