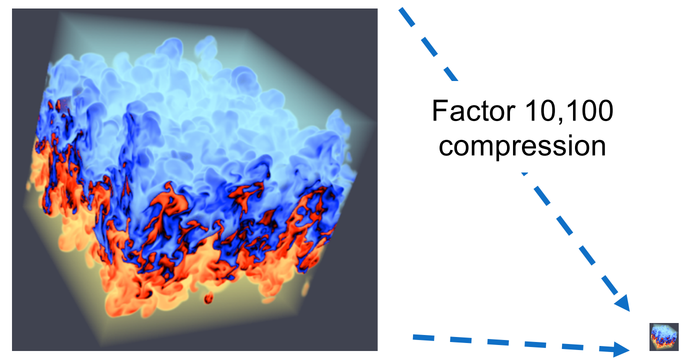
Scientific Data Compression: From Stone-Age to Renaissance

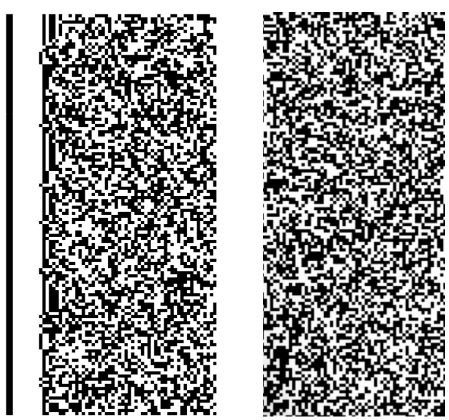
- Background
- Focus on spatial compression
- Best in class lossy compressor
- Open questions



Point wise max error bound: **10^{-5}**

This is what we need to compress (bit map of 128 floating point numbers):

Random noise



Franck Cappello

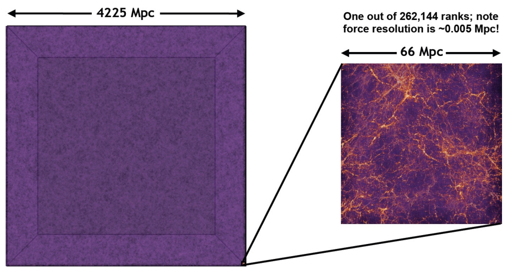
Argonne National Lab and UIUC

CCDSC, October 2016

Why compression?

- Today's scientific research is using simulation or instruments and produces extremely large of data sets to process/analyze

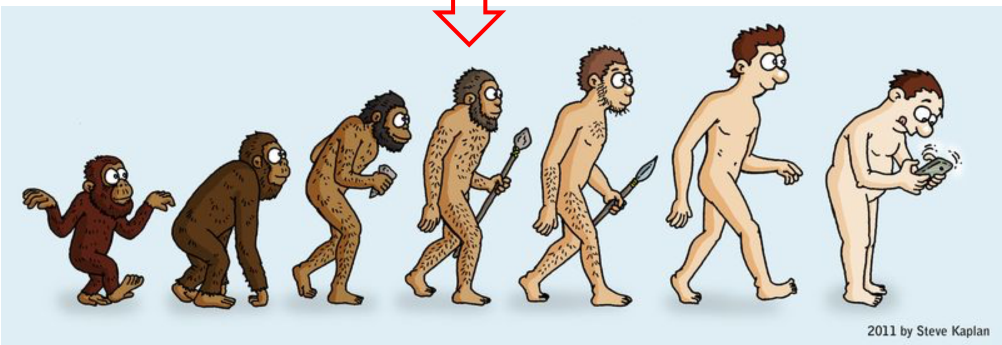
- In some cases, extreme data reduction is needed:
  - Cosmology Simulation (HACC):
    - A total of >**20PB** of data when simulating trillion of particles
    - Petascale systems FS ~20PB (you will never have 20PB of scratch for one application)
    - On Blue Waters (1TB/s file system), it would take 20 X 10^15 / 10^12 seconds (5h30) to store the data → currently drop 9 snapshots over 10
  - Also: HACC uses all the available memory: there is room only for 1 snapshot (so temporal compression would not work)



Stone age of compression for scientific data sets

- Tools were rudimentary
  → Apply compressors developed for integer strings (GZIP, BZIP2) or images (JPEG2000)

10 years ago



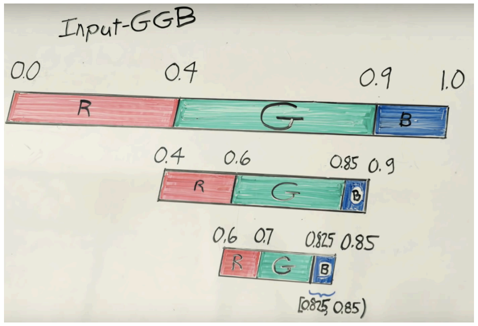
- Tool effects were limited in power and precision
  → Low compression factors
  → First lossy compressors did not control errors

- No clear understanding on how to improve technology
  → Some did not understand the limits of Shannon entropy
  → Metrics were rudimentary: compression factor & speed

- Cultural fear of using lossy compression for data reduction

Artefacts of that period (lossless)

- LZ77: leverages repetition of symbol string
- Variable Length Coding (Huffman for example)
- Move to front encoding
- Arithmetic encoding (symbols are segments of a line [0,1] of length proportional to their probability of occurrence)



- Burrows–Wheeler algorithm (bzip2)
- Markov Chain Compression
- Dynamic Statistical Encoding (adapts dynamically the probability table of symbols for Variable Length Coding)
- *Lorenzo predictor* + *correction*
- Techniques are combined in most powerful compressors: bzip: Burrows–Wheeler + Move to front + Huffman

All these algorithms either leverage string of symbols (bytes) repetition

OR perform probability encoding: variable length coding

Effectiveness of the tools from that period

P. Ratanaworabhan, Jian Ke ; M. Burtscher Cornell Univ., Ithaca, NY, USA
Fast lossless compression of scientific floating-point data
Data Compression Conference (DCC'06) 2006

	bzip2	dfcm	fsd	gzip	lzpx	p7zip	rar	zzip
aztec	1.15	**1.69**	1.42	1.22	1.15	1.39	1.26	1.15
bt	1.10	**1.36**	1.02	1.13	1.10	1.32	1.15	1.10
eulag	1.04	**1.23**	1.06	1.06	1.05	1.15	1.07	1.09
lu	1.02	**1.23**	0.99	1.05	1.03	1.22	1.07	1.03
sp	1.08	1.25	0.95	1.11	1.07	**1.31**	1.14	1.07
sppm	6.78	4.16	2.14	6.31	7.94	**8.31**	7.68	---
sweep3d	1.06	**1.49**	1.20	1.09	1.19	1.26	1.30	1.35
geo_mean	1.40	1.60	1.20	1.42	1.46	**1.66**	1.52	1.13

In SPPM data set, each double value is repeated ~10 times

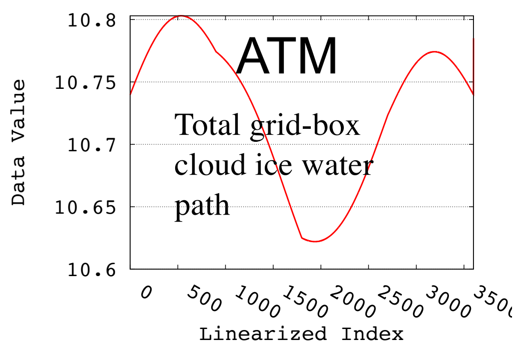
Compression limited to a factor of 2 in most cases

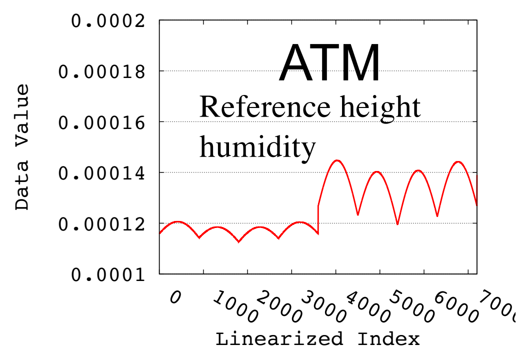
Renaissance: the current period (1)

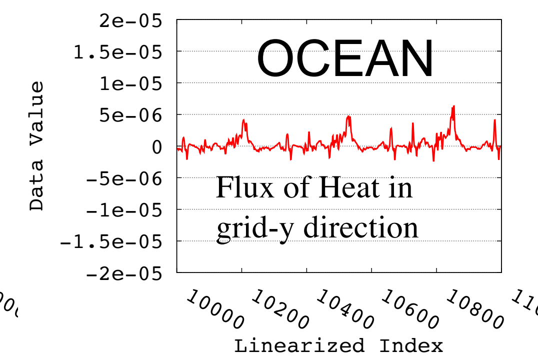
Scientific dataset need specific compressors..
…exploiting their unique properties.

Plotting datasets as time series:

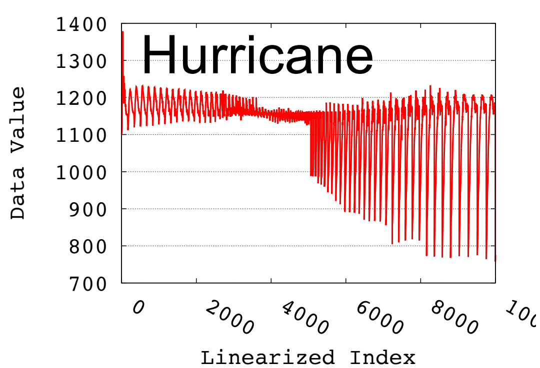
CESM/
ACME

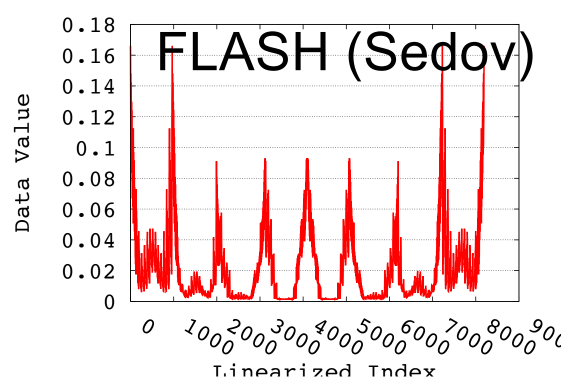


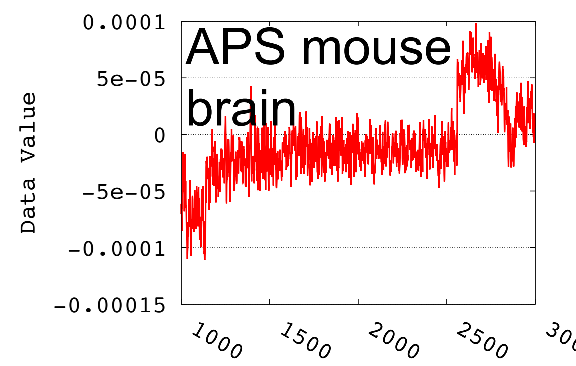




But not all datasets are smooth







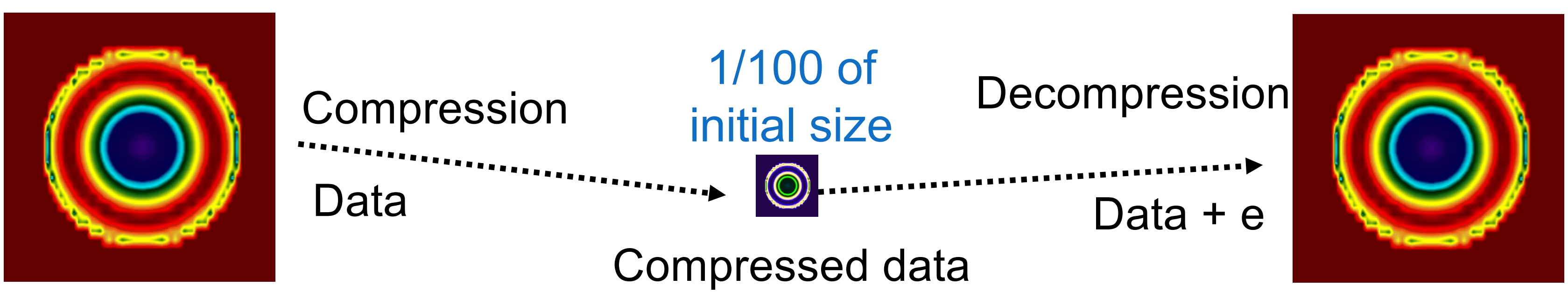
Renaissance (2): Increased acceptance of lossy compression

Argonne NATIONAL LABORATORY

- Tradeoff between data size and data accuracy

- Specific requirements for usefulness:
  - Error-bounded compression: **guaranteeing** the accuracy of the decompressed data for users (multiple metrics).
    → Max error: Typically 10^{-5},
    → PSNR (f(dynamic, mean squared error)) =>100DB (10^{5})

  - **Fast** compression and decompression (if in-situ, compression time should not exceed significantly storage time): x100MB/s on 1 core



Renaissance (3): Explosion of new ideas

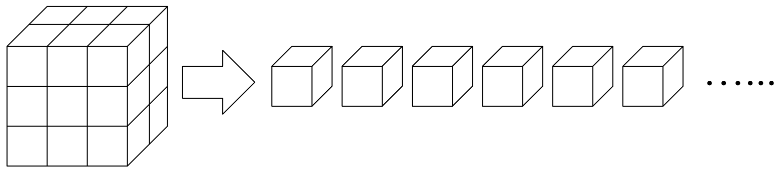
- Lossy compressors
  - ANL/SZ, FPZIP-40, ZFP, ISABELA, SSEM, NUMARCK.
- Common techniques used by related work
  - Vector Quantization (VQ), Transforms (T), Curve-Fitting/Spline interpolation (CFA), Binary Analysis (BA), Lossless compress (Gzip), Sorting (only Isabella), Delta encoding (only NUMARCK), Lorenzo predictor (only FPZIP)

Compressor	VQ	T	CFA	BA	Gzip
NUMARCK [12]	✓				
ISABELA [6]			✓		
ZFP [13]		✓		✓	
SZ [7]			✓	✓	✓
Fpzip [14]				✓	
SSEM [5]	✓	✓			✓

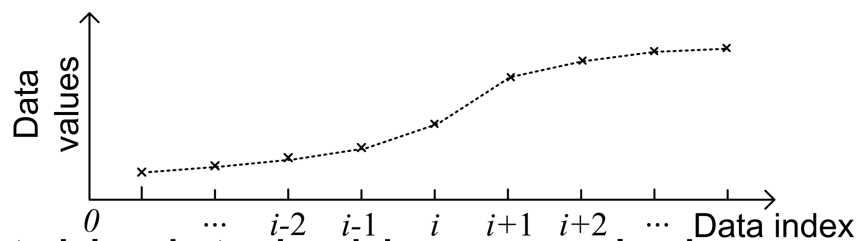
Argonne SZ Best in class compressor for scientific data sets (strictly respecting user set error bounds).

- Basic Idea of SZ 1.1 (four steps):

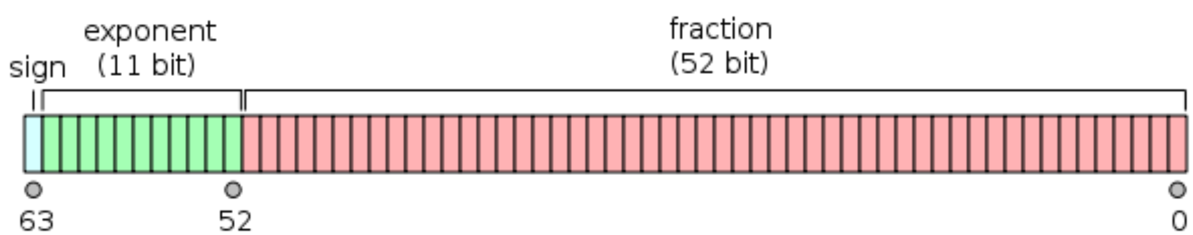
Step 1. Data linearization: Convert N-D data to 1-D sequence



Step 2. Approximate/Predict each data point by the best-fit curve-fitting models



Step 3. Compress unpredictable data by binary analysis

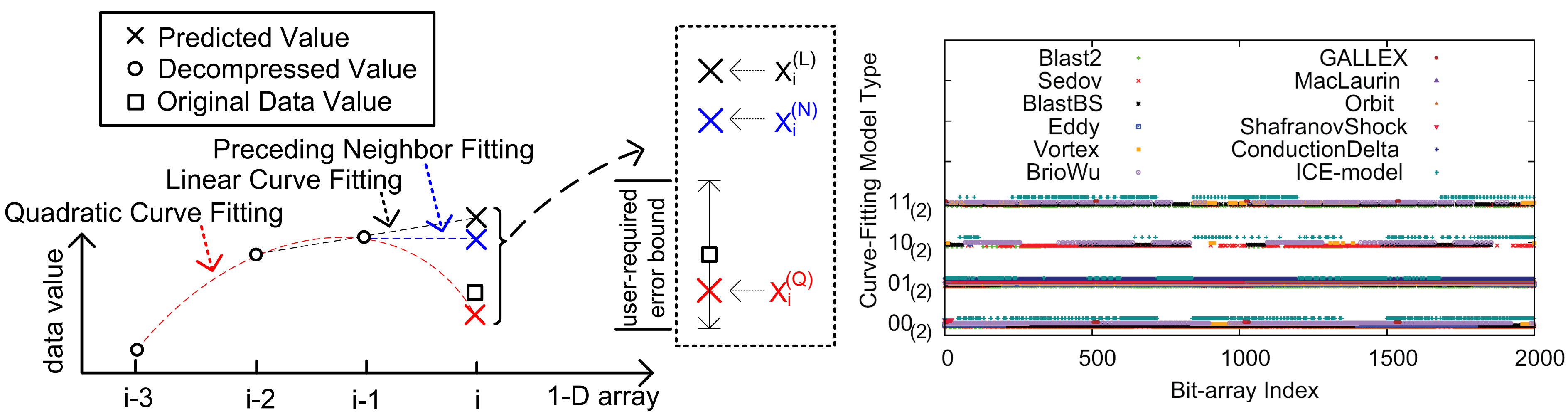


Step 4. Perform lossless compression (Gzip): LZ77, Huffman coding

Steps 1-3 prepare for strong Gzip compression

Step 2 of SZ 1.1: Prediction by best-fit curve fitting model

- Use two-bit code to denote the best-fit curve-fitting model
- 01: Preceding Neighbor Fitting (PNF)
- 10: Linear Curve Fitting (LCF)
- 11: Quadratic Curve Fitting (QCF)
- 00: This value cannot be predicted – <u>unpredictable data</u>



SZ 1.1 Error control

- Two types of error bounds are supported
  - **Absolute Error Bound**

    Specify the max compression error by a constant, such as 10^{-6}
  - **Relative Error Bound**

    Specify the max compression error based on the <u>global value range</u> size and a percentage

Combination of Error Bounds

Users can set the real compression error bound based on only absErrorBound, relBoundRatio, or a kind of combination of them. Two types of combinations are provided: **AND, OR**.

The combined error bound is then computed by the **Min** of the two error bounds (AND) or the **Max** (OR)

Evaluation Results

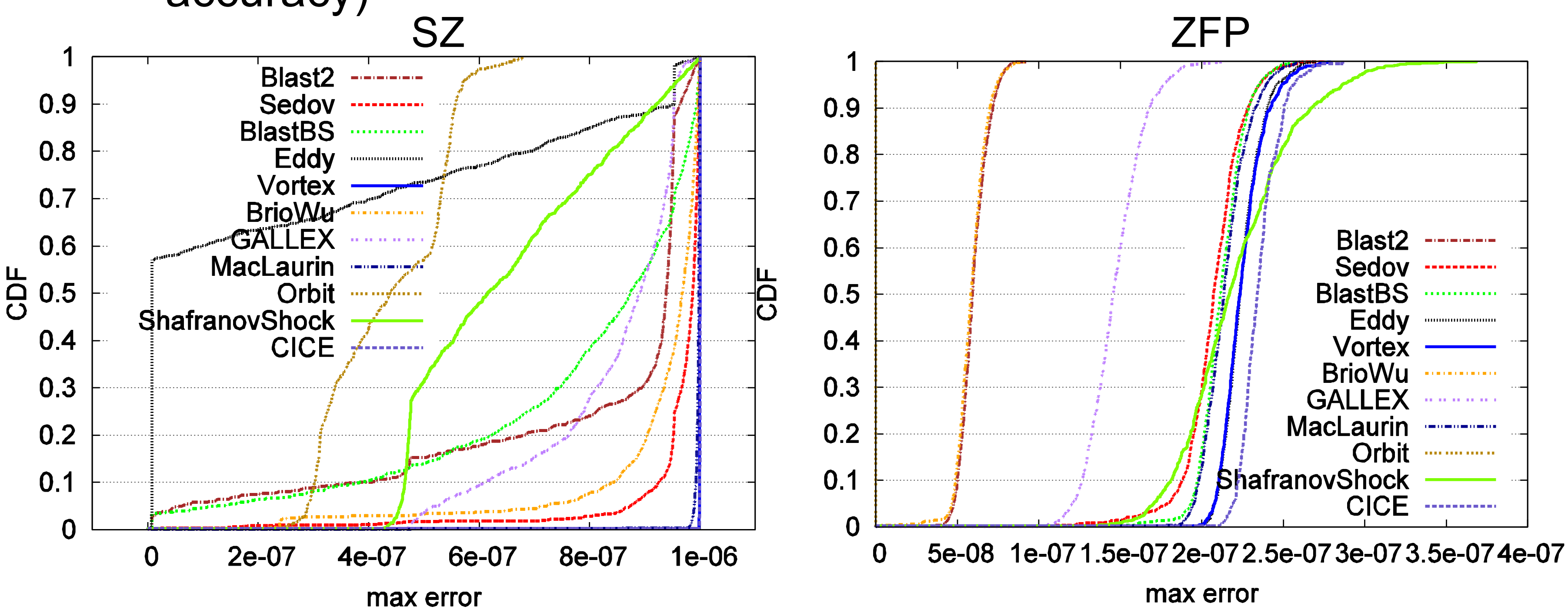
- Compression Factor (EB: 10^{-6}): original size / compressed

Benchmark	$SZ_{1.1}$	ZFP	ISA	SSEM[a]	FPZIP-40[b]	Gzip	FPC[c]
Blast2	110	6.8	4.56	39.7	22.9	77	11.4
Sedov	7.44	4.75	4.42	17[d]	3.43	3.13	1.9
BlastBS	3.26	3.65	4.43	8.45	2.43	1.24	1.29
Eddy	8.13	8.96	4.34	N/A	2.56	5.5	3.89
Vortex	13.6	10.9	4.43	12	3.35	2.23	2.34
BrioWu	71.2	8.24	5	35.7	21.9	73	8.5
GALLEX	183.6	36.7	4.89	82.4	20.35	34.7	11.37
MacLaurin	116	21.77	4.1	7.44	3.84	2.03	2.08
Orbit	433	85	4.96	11.7	3.9	1.8	1.86
ShafranovShock	48	4.43	4.24	20.3	19.9	28	7.33
CICE	5.43	3.52	4.19	3.83	2.3	2.6	2.67
ATM	3.95	3.17	3.1	1.82	1.04	1.36	N/A
Hurricane	1.63	1.19	2.57	1.11	2.07	1.16	N/A

- SZ 1.1 Compression Factor > 10 for 7 of the 13 benchmarks
- SZ 1.1 better than ZFP for all datasets but 2

Evaluation Results

- Compression Error
  - Cumulative Distribution Function over the snapshots
  - SZ and ZFP can both respect the absolute error bound 10^{-6} well.
  - SZ is much closer to the error bound (ZFP over preserves data accuracy)



However, in some situations ZFP does not respect the error bound (observed on the ATM dataset from NCAR)

Evaluation Results

- Compression Time (in seconds)

High cost due to sorting operations

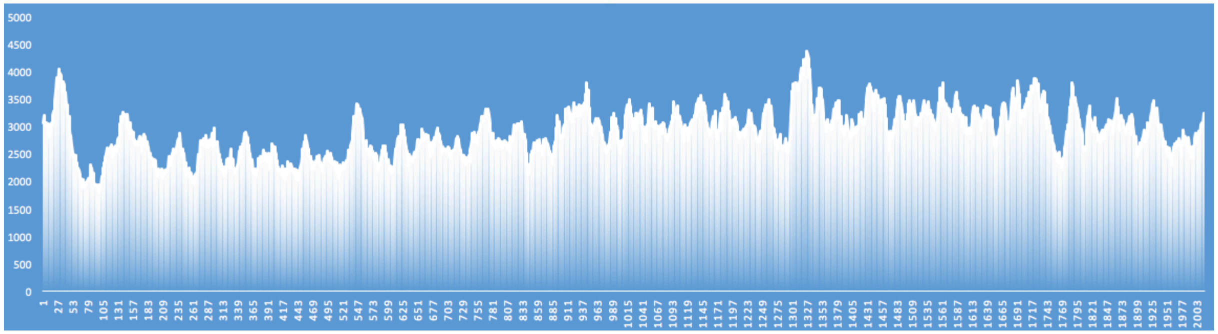
SZ 1.1 compression time is comparable to ZFP

Application	DataSize	ISA.	ZFP	SZ w/o Gzip	SZ with Gzip
Blast2	787MB	129	8	6.4	9.4
Sedov	660MB	115	9.3	6.3	11.9
BlastBS	984MB	73.2	17	11.9	23.1
Eddy	820MB	143	17.4	8	14.2
Vortex	580MB	108	8.6	5.5	8.7
BrioWu	1.1GB	132	9.6	8.7	9.8
GALLEX	270MB	31	1	1.9	2.5
MacLaurin	6.3GB	1285	55	22.8	28.5
Orbit	152MB	19	0.7	0.56	0.95
Shaf.Shock	246MB	38.3	4.9	1.7	2.9
Cond.Delta	787MB	84	6.2	3.8	6.2
CICE	3.7GB	790	90.2	39.8	84.3
ATM	1.5TB	-	25604	24121	38680
Hurricane	4.8GB	1152	155	156	237

More research is needed (1)

Some datasets are “hard to compress”

- All compressors (including SZ) fail to reach high compression factors on several data sets:
  - BlastBS (*3.65*), CICE (5.43), ATM (3.95), Hurricane (1.63)
  - We call these data sets “hard to compress”
- A common feature of these datasets is the presence of spikes
  - If you plot the dataset as a time series:
    - Example:
    - APS data (Argonne photon source)

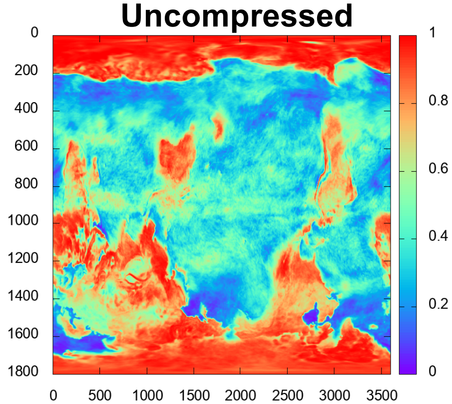


More research is needed (2):
What are the right metrics?

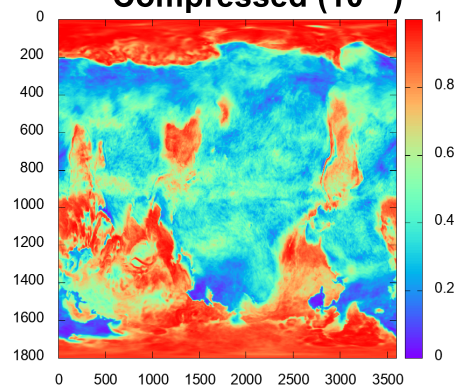
Argonne NATIONAL LABORATORY

Variable FREQSH (Fractional occurrence of shallow convection) **in ATM Data Sets (CESM/CAM)**

Uncompressed



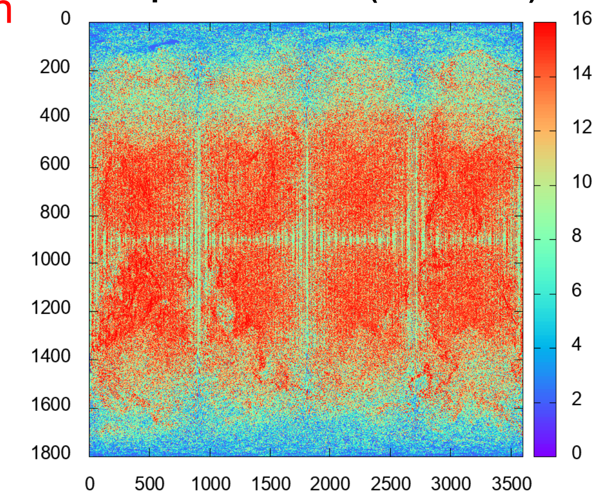
Compressed (10^{-4})



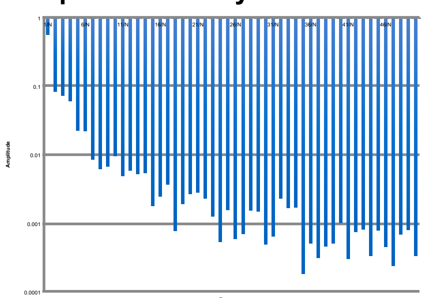
SZ Compression factor: 6.4

(1.4 with GZIP)

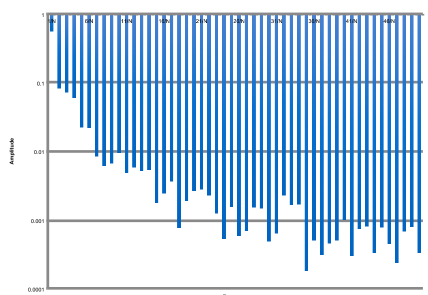
Compression rate (bits/value)



Spectral density estimation



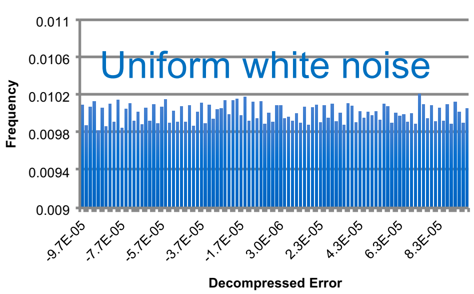
Spectral density estimation



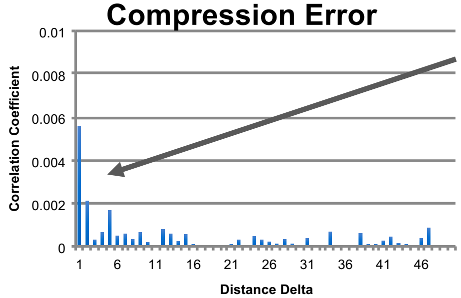
Laplacian comparison

(original versus compressed)

Distribution of Decompression Error

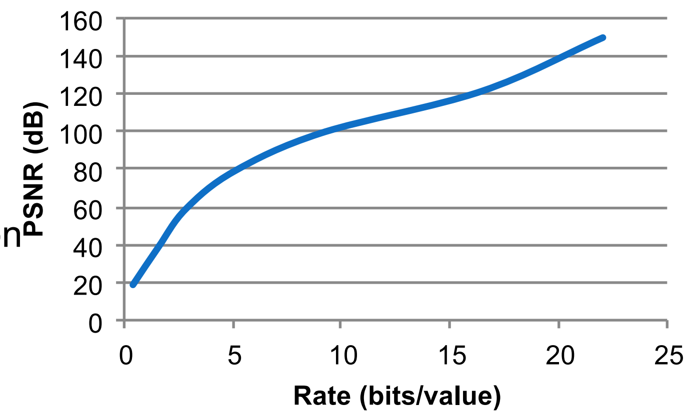


Autocorrelation of Compression Error



Compression adds extra correlation

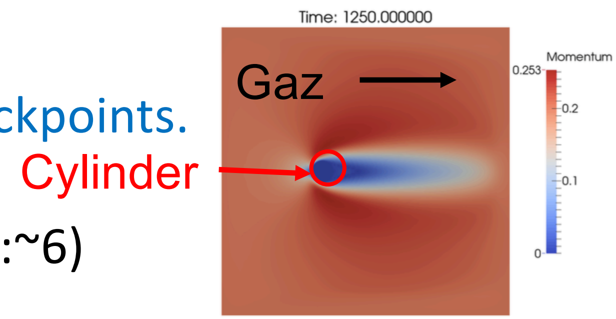
Rate-Distortion

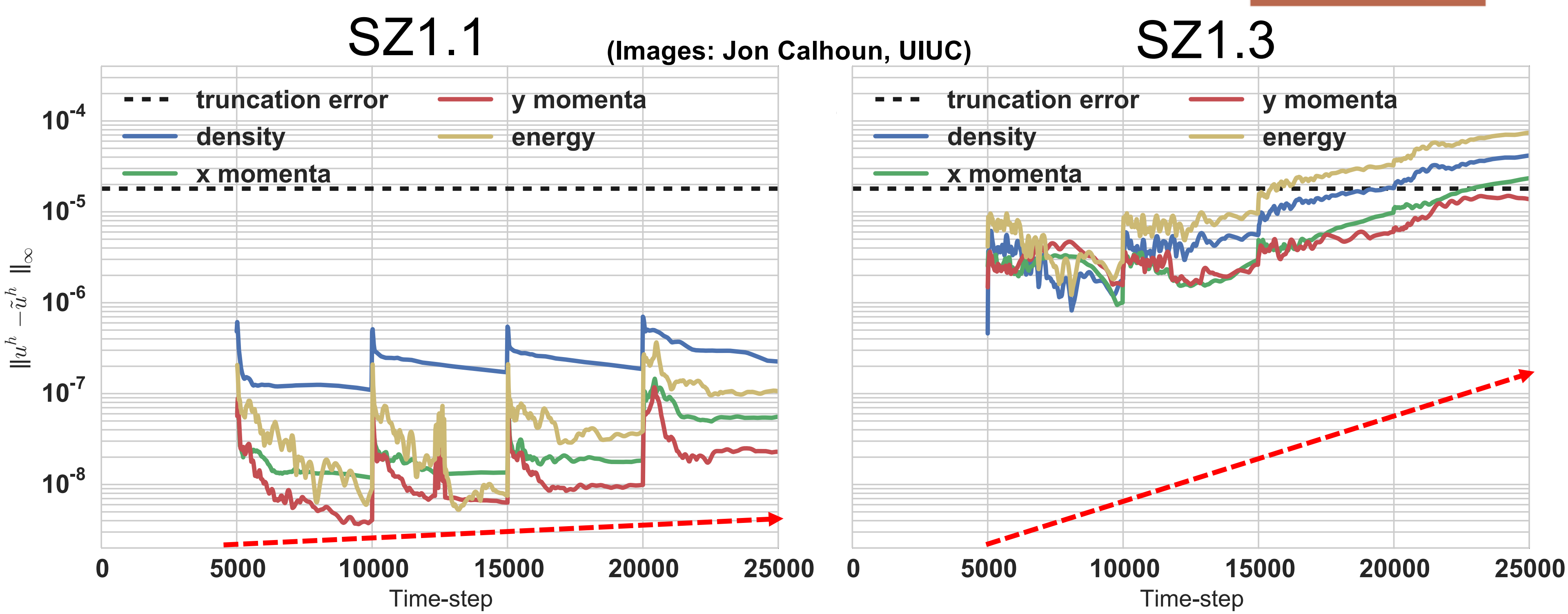


More research is needed (3):

Respecting error bound does not guarantee temporal behavior

- PlasComCM: coupled multi-physics plasma combustion code (UIUC) solving compressible Navier-Stokes equations.
- Truncation error is at 10^{-5}
- We checkpoint it and restart from lossy (EB=10^{-5}) checkpoints.
- We measure derivation from lossless restarts
- Two different algorithms SZ 1.1 (CF:~5) and SZ 1.3 (CF:~6)





(Images: Jon Calhoun, UIUC)

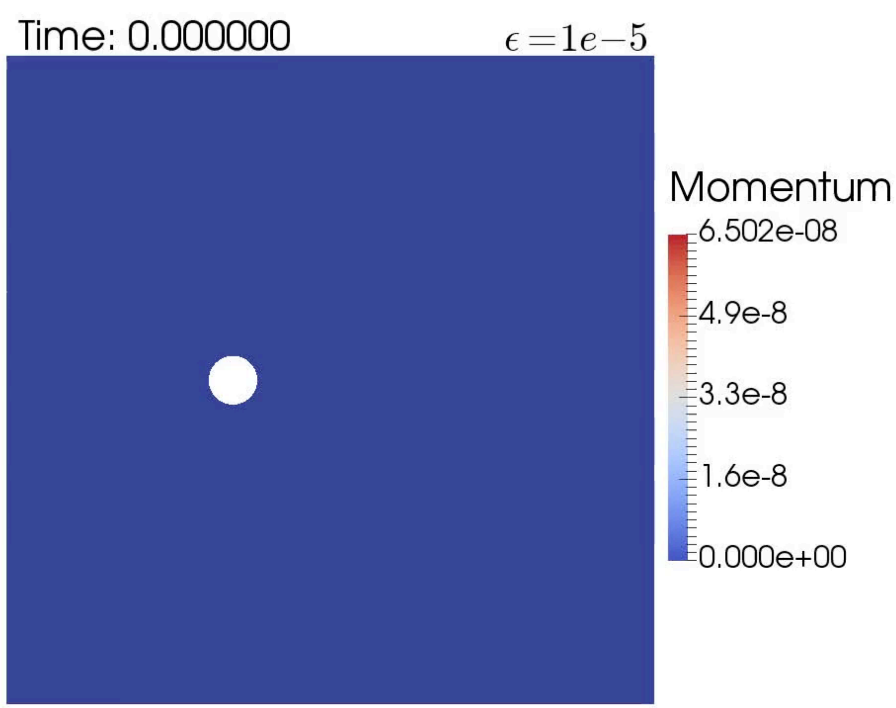
More research is needed (4):

Respecting error bound does not guarantee spatial behavior

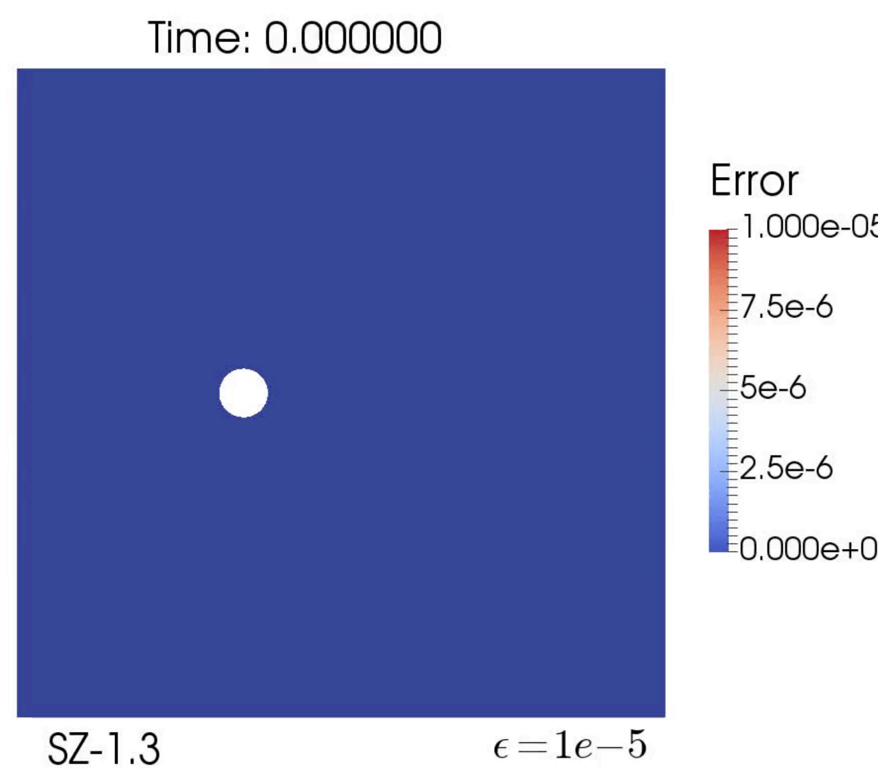
Maximal absolute error between the numerical solution of momentum and the compressed numerical solution of momentum in PlasComCM.

(Images: Jon Calhoun, UIUC)

SZ1.1



SZ1.3



Conclusion

- The world of compression is fascinating!
  - This is just the beginning.
- There is still a lot to be done:
  - ”Hard to compress data sets”
  - Identify relevant compression metrics
  - Understand/control propagation of compression error
  - Opportunities for co-design
- Preparing a workshop at Argonne in March 2017 ”Lossy compression for scientific computing and data analytics”
- If you interested, send me an email.

By the way, compression is also an art