# High Performance Scientific Computing - from Simulation to Optimisation

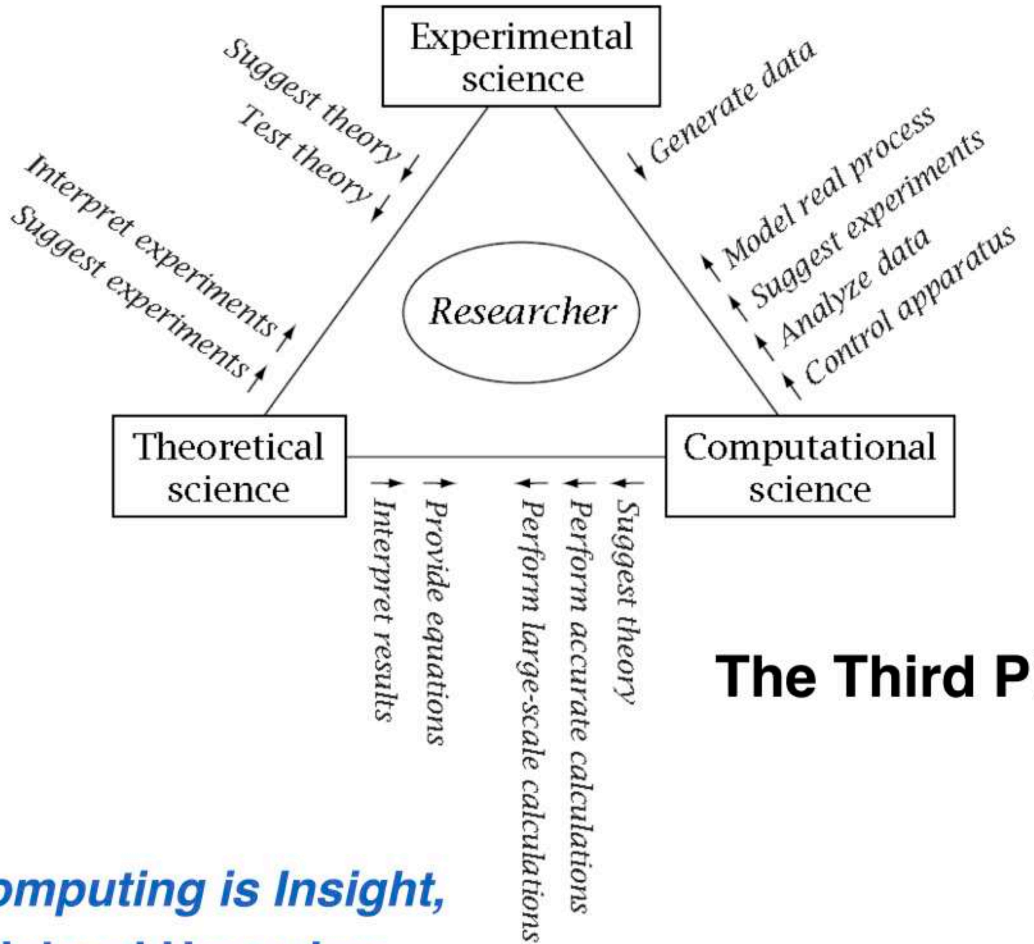
Emil-Ioan Slușanschi

Smart Diaspora 2023 - Bioinformatics without frontiers: from infrastructure to multidisciplinary applications

Timisoara - 11.04.2023

University Politehnica of Bucharest

# Why HPSC?

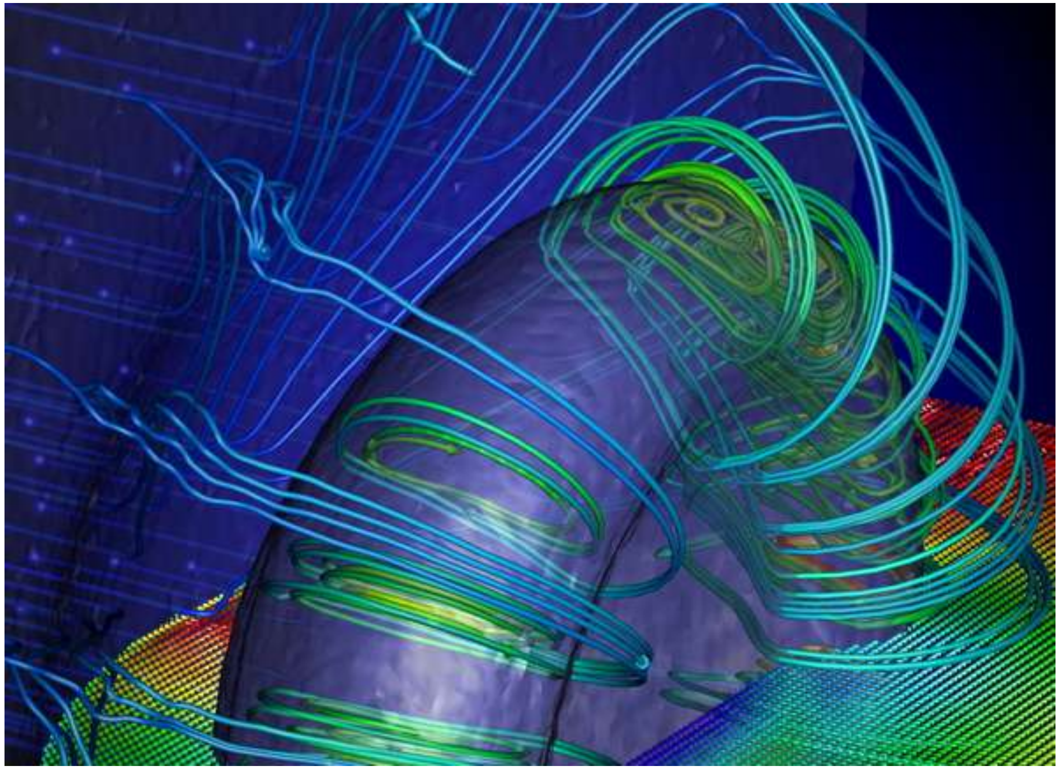



**The Third Pillar of Science**

***"The Purpose of Computing is Insight, not Numbers!"*** **- Richard Hamming**

# Scientific Computing

- Computational Modeling:
  - ODE / PDEs
  - Simulations: particle simulations, simulated annealing
  - Cellular automata
  - Statistical models
  - Suitable algorithms & data structures
- Software for (parallel) computers



# Paradigm Shift

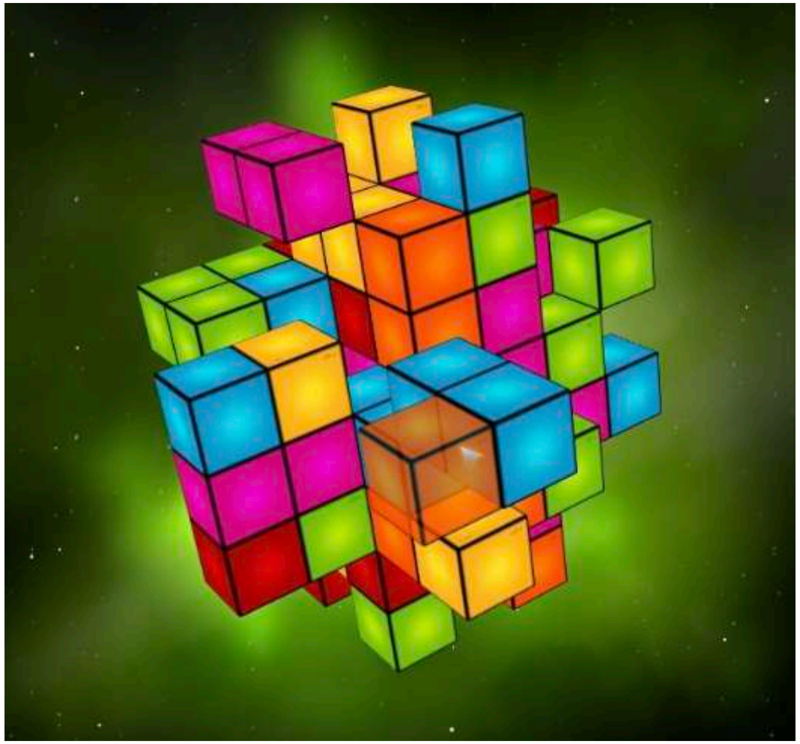

- The 20th Century was limited by what we could measure
- The 21st Century will be limited by what we can compute
  - See farther
  - Move through time
  - Verify difficult hypotheses
  - Safely conduct experiments

# Will We ever have “Enough"?

- 2-3x is the Free Lunch
- 5-10x is worth Upgrading
- 100x+ significantly changes Time-to-Discovery
  - Code Redesigning
  - New HW/SW Platform
- Never enough Computational Power

**IS NEVER ENOUGH**

# Critical Mass



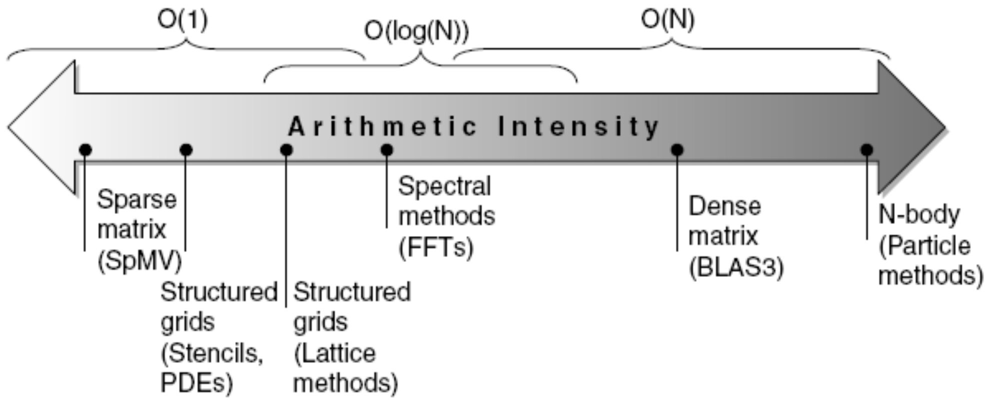
- Computational Experimentation
  - Large systems
  - Long simulations
  - Enough details
- Computational Instrumentation
  - Good accuracy
  - Enough observations

# Will We be able to Scale Real Apps?

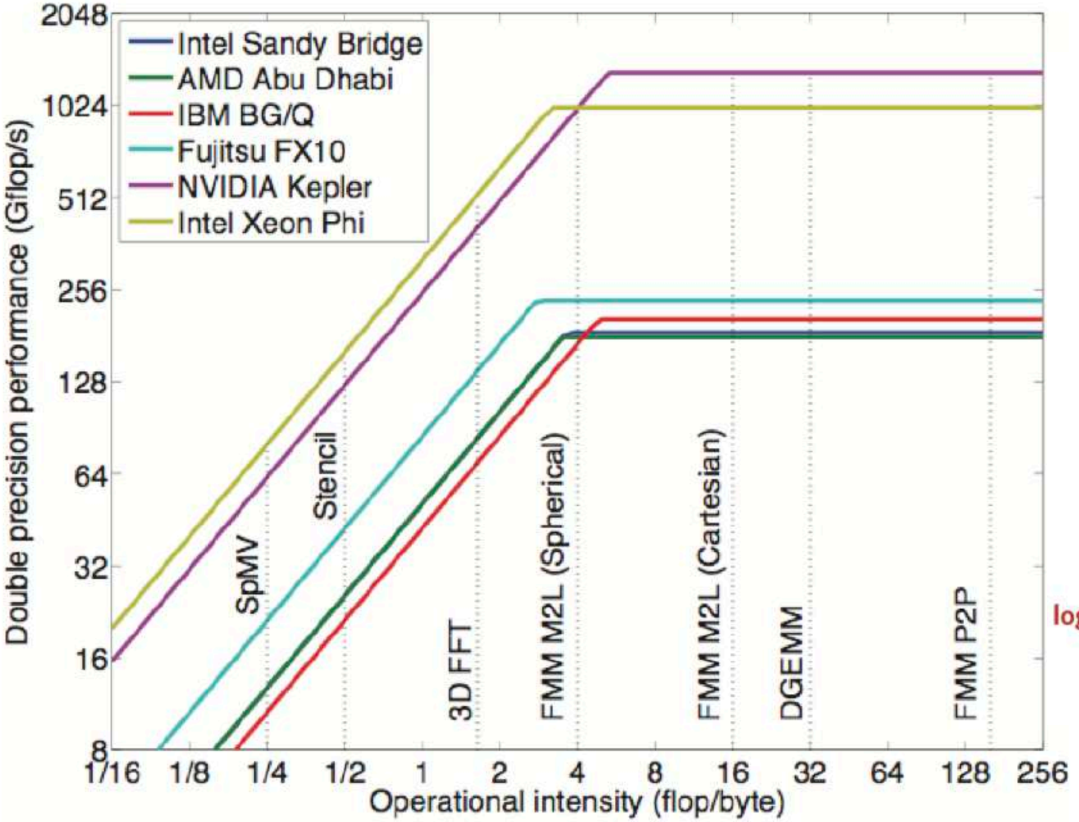
- HPC is a moving target
- The rate of growth of HW has been exponential … so far
- However, most workloads don't scale!
  - TB to PB to EB…
- Parallel tasks != parallel computing…
- We do have a… SW scaling problem!

# The Roofline Performance Model

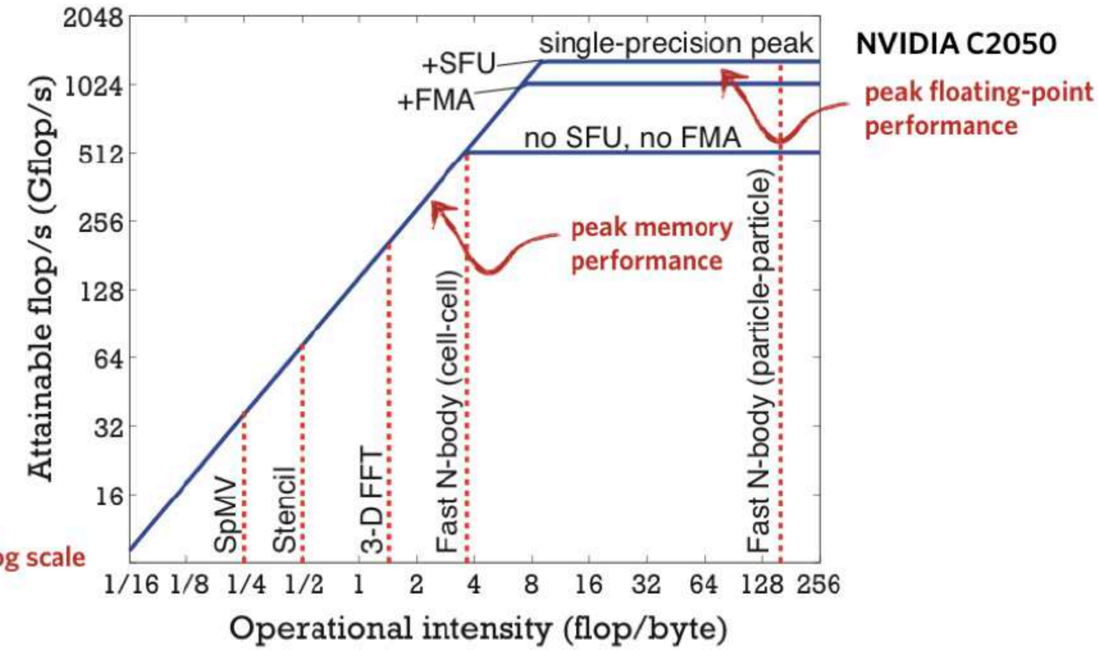
- Basic Idea
  - Plot peak floating-point throughput as a function of arithmetic intensity
  - Ties together floating-point performance and memory performance for a target machine
- Arithmetic Intensity
  - Floating-point operations per byte read



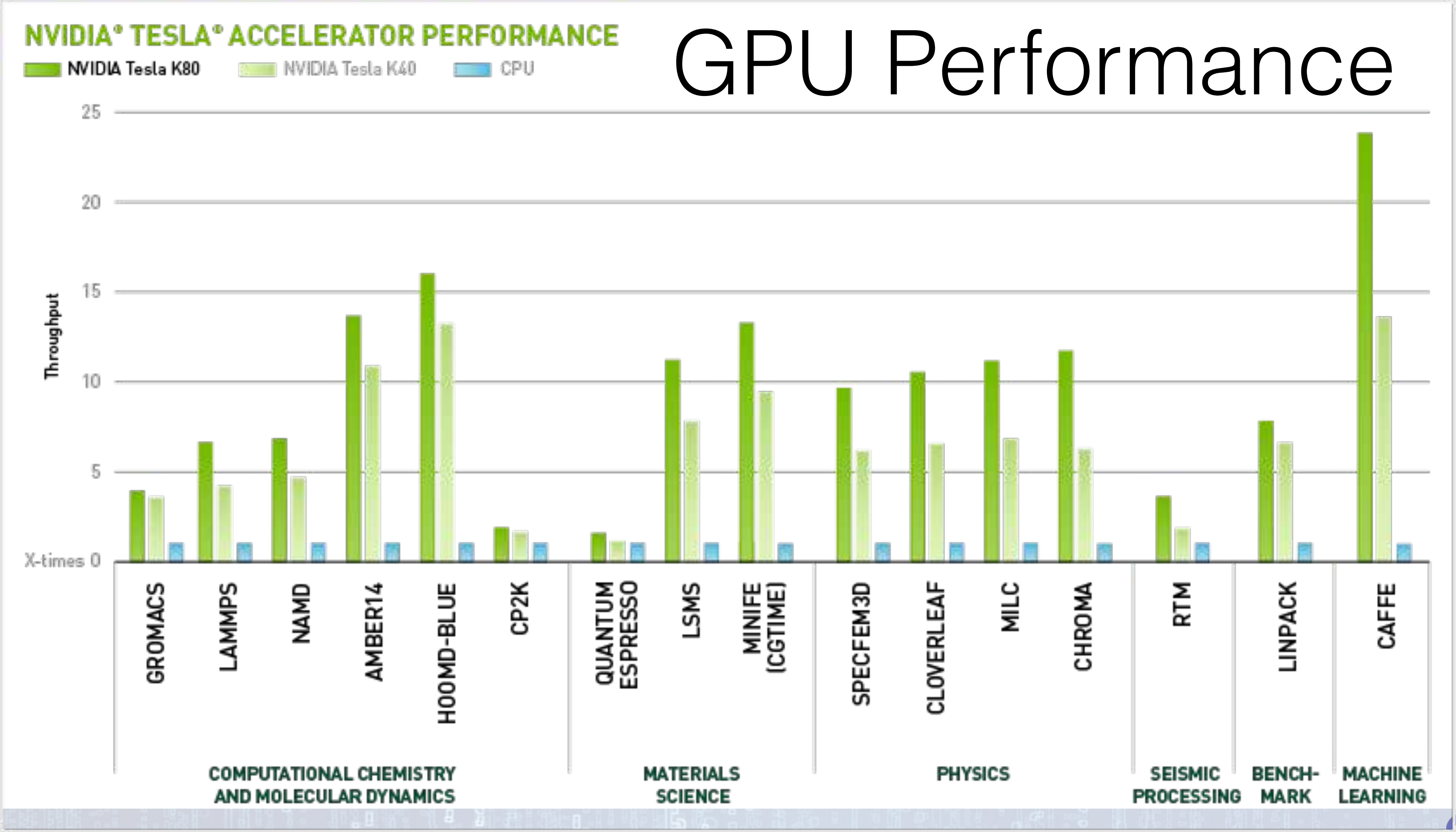
# Roofline Performance Examples



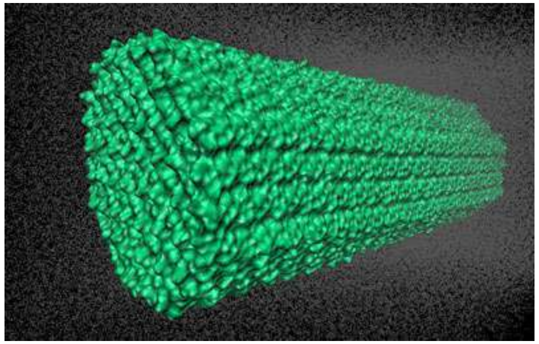

log/log scale



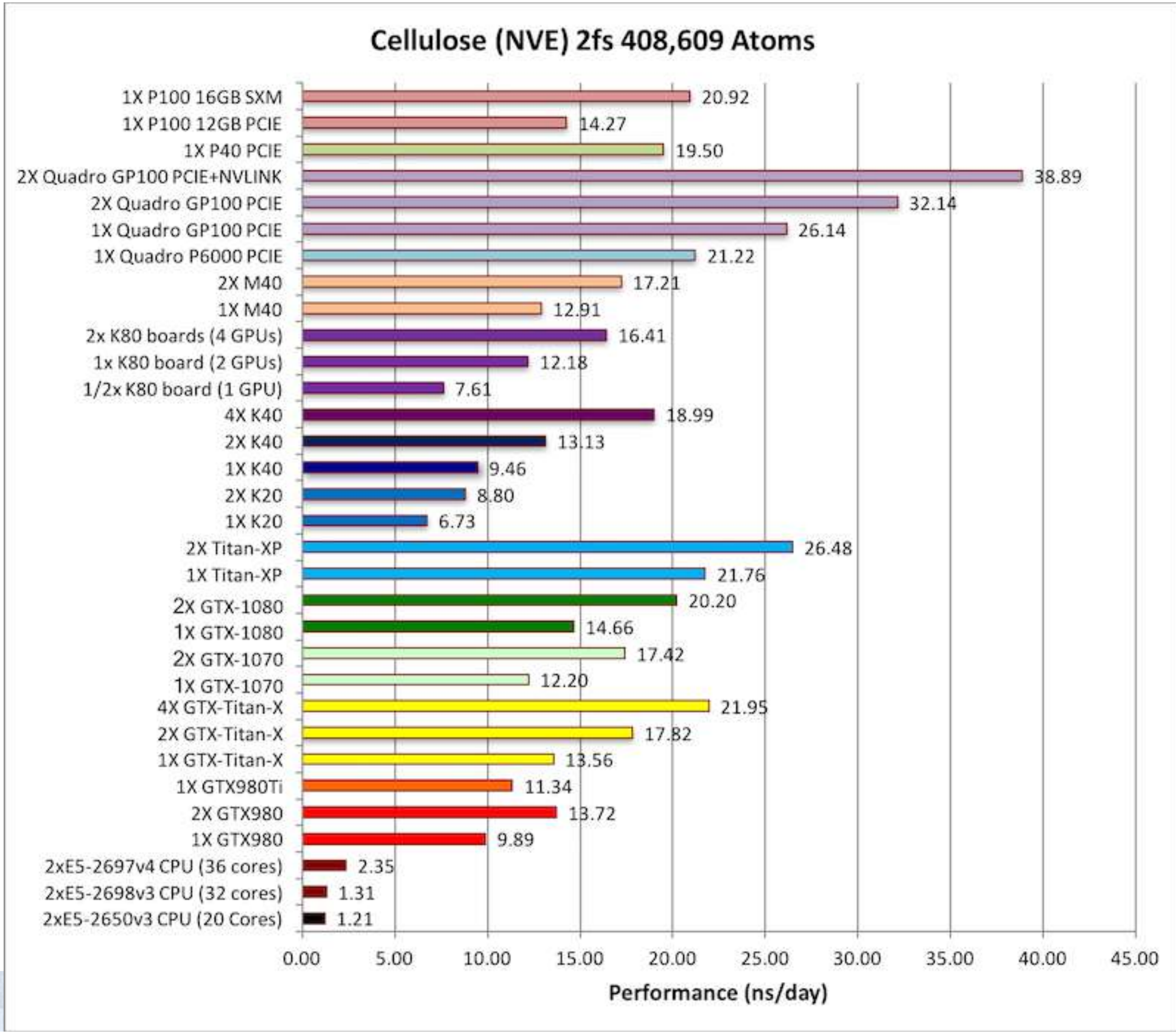
# GPU Performance

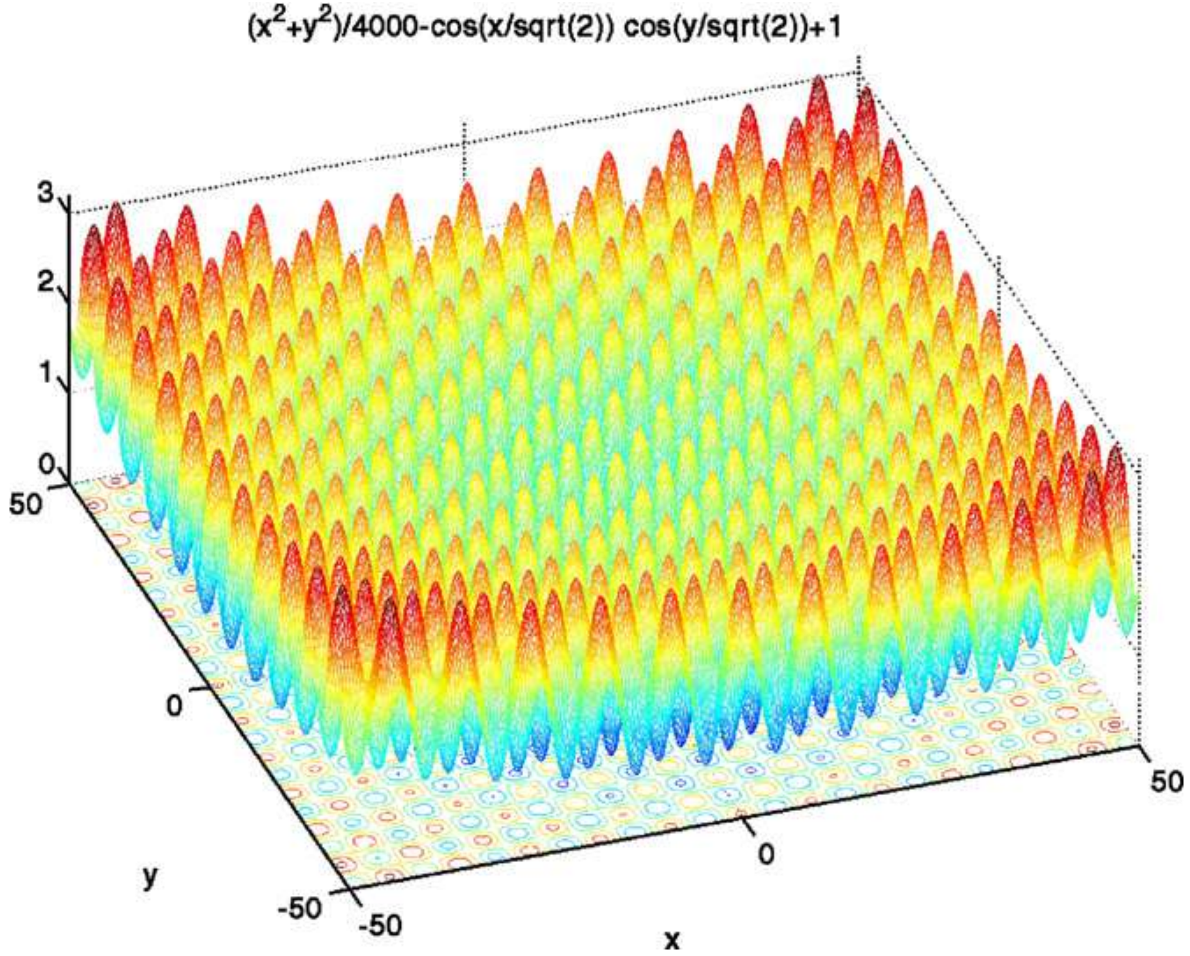


# Amber Molecular Simulations



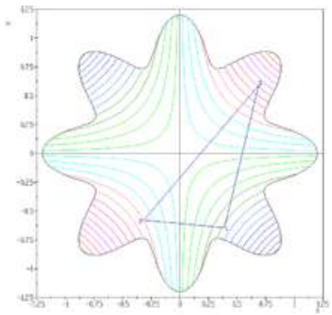
Cellulose






# Numerical Optimisation

# Optimisation Algorithms



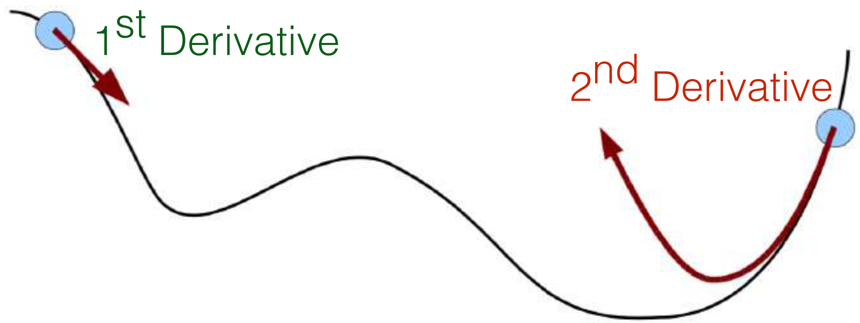
- Gradient Based:
  - Requires 1st / higher order derivatives
  - Guarantees finding of local minima in a small(er) number of iterations
    - First derivative methods typically require $N^2$ iterations
    - Hessian methods typically require N (more expensive) iterations
- Non-Gradient Based:
  - Use only function evaluations
  - May find global minimum / requires large number of iterations
  - Able to find “almost” optimum for non-smooth 1st/2nd derivable functions
  - Examples: Genetic algorithms, grid search, stochastic, etc.

# Gradient or Non-Gradient Based?

- The best approach is in two stages:
  - A low-precision **solver** with Non-Gradient-Based **optimisation** method in the Conceptual Design Stage
    - It gets you close to a global minima
  - A high-precision **solver** with a Gradient-Based **optimisation** method to refine the solution / design
    - It starts close to the global minima and converges on it
- The hard problem:
  - Figure the "proper" combination of **solver** & **optimisation algorithm**

# Is Your Function…

- Differentiable:
  - Gradients indicate the general position of the closest local minima
  - Hessians can offer an estimation of the position of the closest local minima




- Non-Differentiable:
  - No such intuition / hints if derivates are unavailable or too costly to compute

# Derivatives

- Design optimization (single/multiple-objective)
- Sensitivity analysis
- Parameter estimation & fitting
- Data assimilation problems
- Inverse problems…

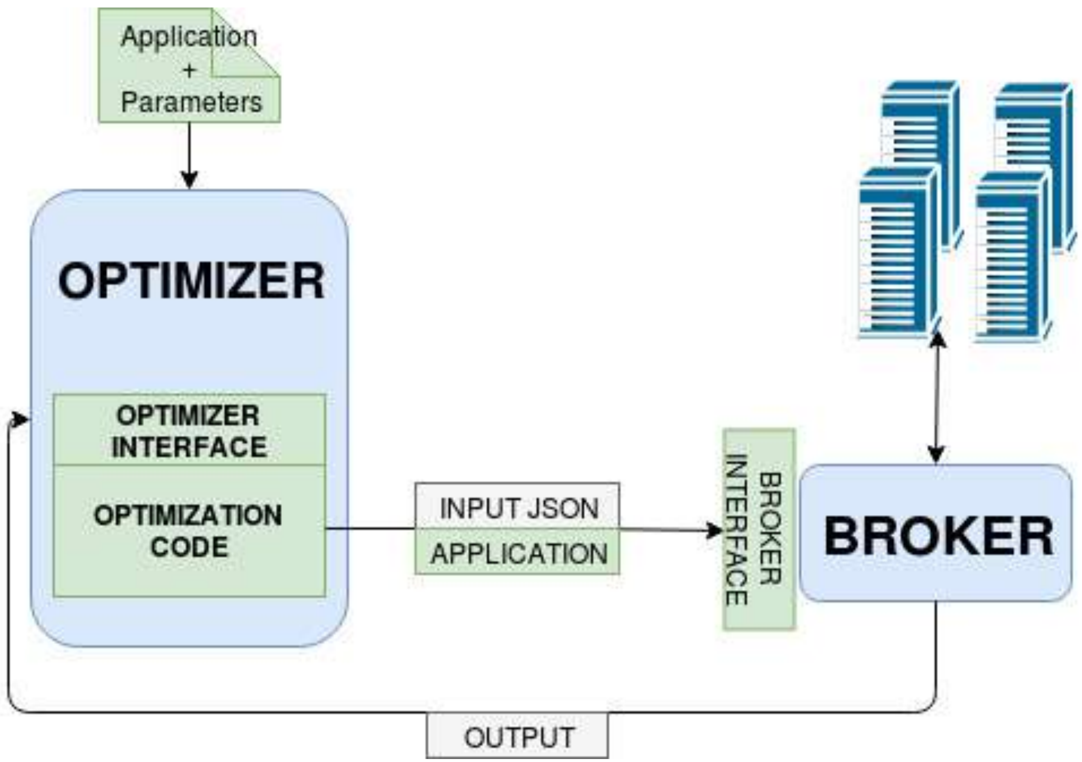
**Derivative Information Required**

Numeric Differentiation

Symbolic Differentiation

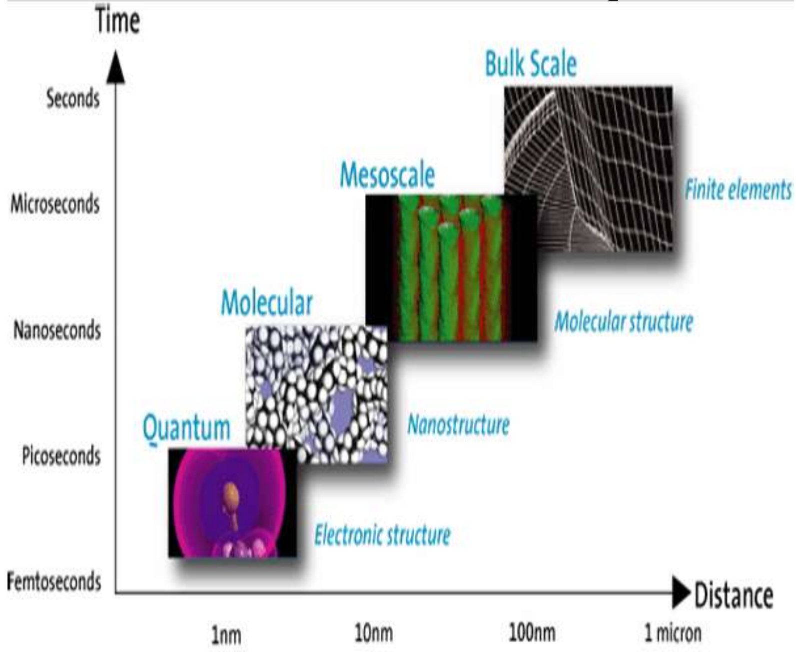
**Automatic Differentiation**

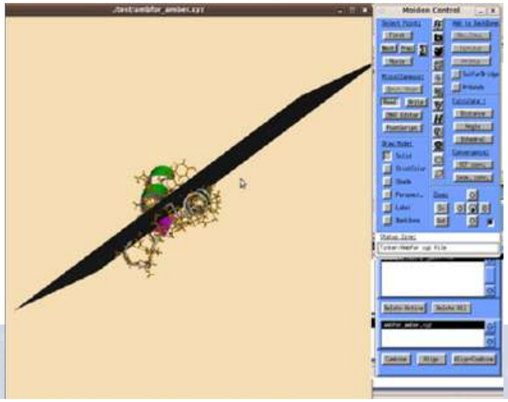
# COSMOS: Framework for **CO**mbining **S**i**M**ulation and **O**ptimization **S**oftware



# Computational Chemistry

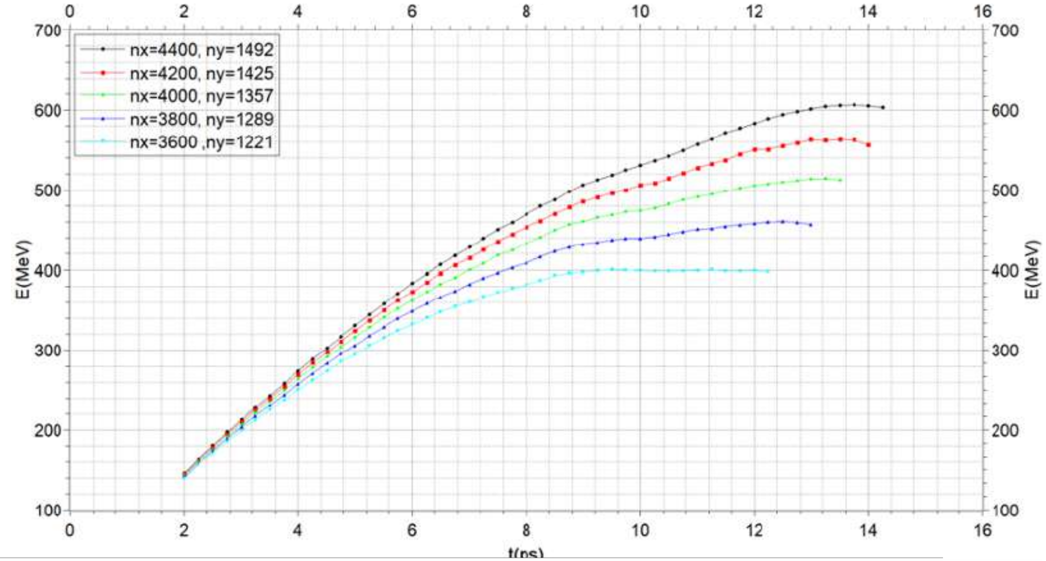
- Paramagnetic materials simulation
  - Shared memory superlinear speedup due to improved cache performance
- GAMESS
  - Distributed Memory Parallelisation
- MOPAC
  - Parallelization schemes for modern multi-core architectures
- NAMD / GROMACS / Gaussian / CPMD / VASP
  - Scale to production systems with CPUs/GPUs

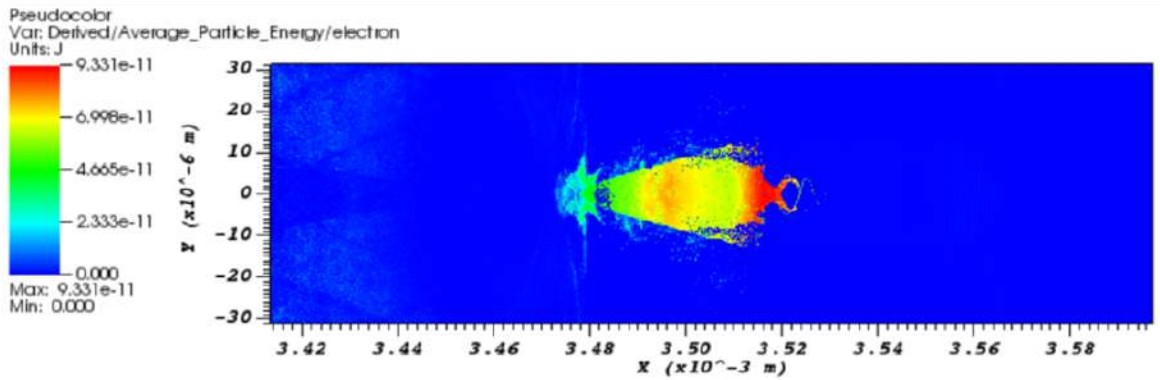


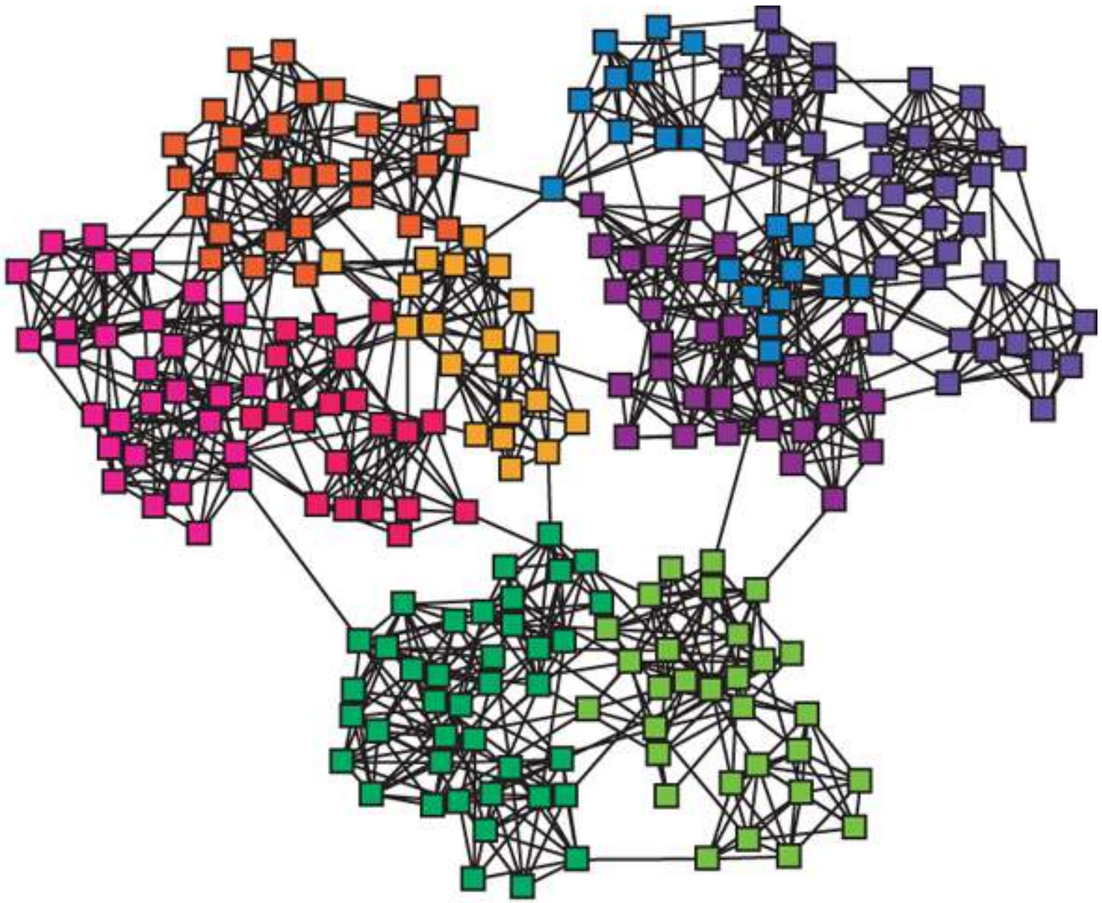
# Optimization of PIC simulations for LWFA

- PIC Simulation
    - Energy distribution at 12ps
- Grid independence study
- Optimize the characteristics of the simulation to match the Laser Experiments

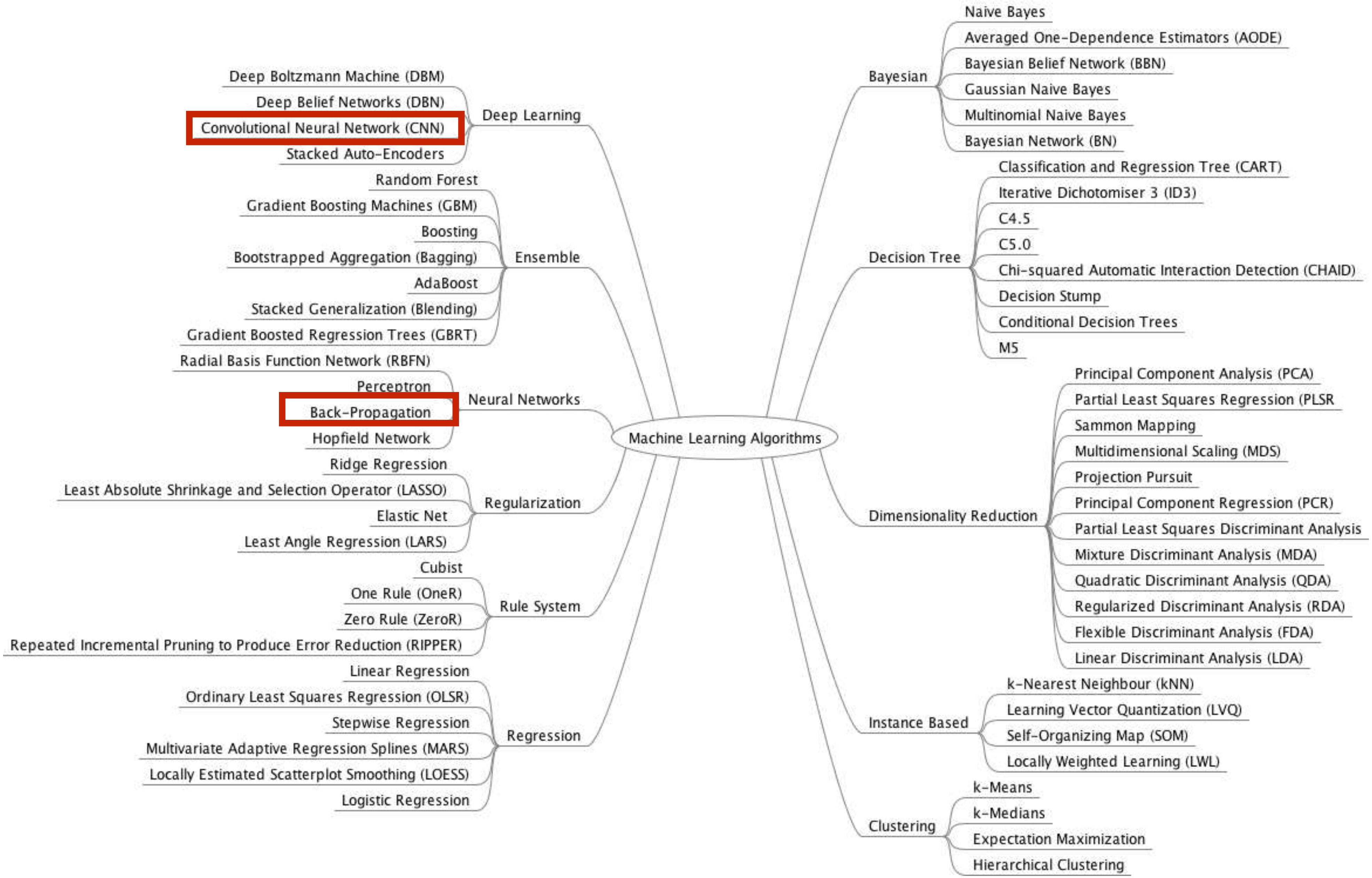


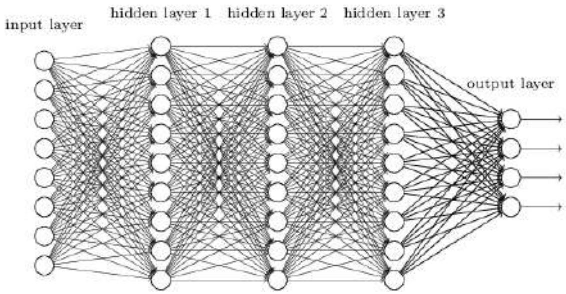

nuclear physics



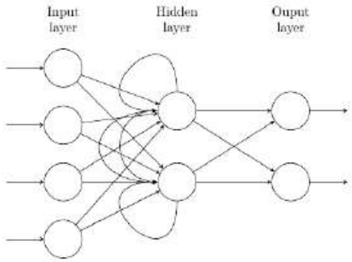
# Machine Learning Tech


Machine Learning Algorithms
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering

# Differentiability & end-to-end Training




Deep CNN




RNN

Numerous people in an indoor market.

Lots of fruits on the stands.

- Even in complicated scenarios **differentiability** ensures NN can be **trained end-to-end** with back-propagation

# DNN Problems

- While providing good/decent performance DNNs
    - Are used and treated as “**black-boxes**”
    - Lack decomposability into intuitive/understandable components
    - When failing, it is **almost impossible to know/find the reason**
    - **Can’t build trust** without understanding why they predict what they predict
    - **Can’t offer insights** / tradeoffs on stronger/weaker networks
    - Train on **limited data** and testing / working on **real-world data**: **bias / overfitting**

# Going Forward

# Closing the Software Gap

- HPC programming aims performance
- In the future, code productivity is much more important
  - Correct
  - Efficient use of HPC resources
  - Maintenance over a series of architectures
- Moore‘s law gives us freedom to choose architectures…

# HW/SW Challenges

- Power & energy per operation of
  - Computation
  - Data transport
  - Memory
- Threading software to (at least) millions threads
- Address memory and storage capacity & bandwidth limitations
- Managing high-node count systems in the existence of MTBF
- Affordability

# Conclusions

- Hardware/Software systems become the key for innovation in Science
  - Variety of HW-specific architectures
  - Exploiting the resulting freedom of choice software becomes ever more important in the algorithms-software-hardware codesign triad
- Sensible use of HPSC for real applications requires:
  - Algorithmic innovations in applied sciences and numerics
  - New paradigms for software development & infrastructure support
  - Adaptation / tuning /development of appropriate optimisation strategies

# Acknowledgements


- **ANM** – Agentia Nationala de Meteorologie: Rodica Dumitrache, Cosmin Barbu, Doina Banciu, Victor Pescaru
- **IFIN-HH / ISS INCD** – Fizica si Inginerie Nucleara “Horia Hulubei”:
  - Alexandru Nicolin, Mihnea Dulea, Octavian Carbunar, Ionut Vasile, Dragos Ciobanu-Zabet, Gina Isar
- **INFP** – Institutul National de Fizica Pamantului: Mircea Radulian, Constantin Ionescu
- **Unibuc** – Universitatea din Bucuresti: Virgil Baran, Marian Ivan, Gh. Stefanescu
- **UTCB** – Universitatea Tehnica de Constructii Bucuresti: Alexandru Aldea, Cristian Ghindea
- **INCAS** – Institutul National de Cercetari Aerospatiale: Catalin Nae, Victor Mihai Pricop, Marius Gabriel Cojocaru, Claudiu Vadean
- **ICF** – Institutul de Chimie Fizica al Academiei Romane: Viorel Chihaia, Gabriel Munteanu
- **AIRA** – Institutul Astronomic al Academiei Romane: Marian Suran, Dumitru Pricopi
- **UPB-IA** – Universitatea Politehnica Bucuresti, Facultatea de Inginerie Aerospatiala: Marius Stoia-Djeska
- **TU Darmstadt** – Technische Universität Darmstadt: Christian H. Bischof, Andreas Wolf
- **GA Uni Goettingen** – Georg-August Universität Göttingen: Johannes Dieterich, Jonas Feldt
- **Uni Jena** – Friedrich-Schiller-Universität Jena: Martin H. Buecker, Martin Engler
- **INRIA** – **Inria Sophia Antipolis**: Laurent Hascoët
- **UPB-CS** – **Universitatea Politehnica Bucuresti**, Facultatea de Automatica si Calculatoare:
  - Nicolae Tapus, Alexandru Herisanu, Voichita Iancu, Silvia Stegaru, Dan Tudose, Razvan Dobre, Ovidiu Hupca, Maria Nadejde, Cosmin Constantin, Marius Poke, Vlad Spoiala, Cristina Ilie, Victor Spiridon, Ana Sima, Valentin Marcu, Vlad Dumitrel, Daniel Mahu, Alex Teaca, Cosmin Samoila, Rareș Folea, Radu Daia, Mariana Ioniță, Sebastian Muraru

Thank you for your attention.

emil.slusanschi@cs.pub.ro
cs.pub.ro