中国科学院大学

2026

数值代数

Numerical Algebra

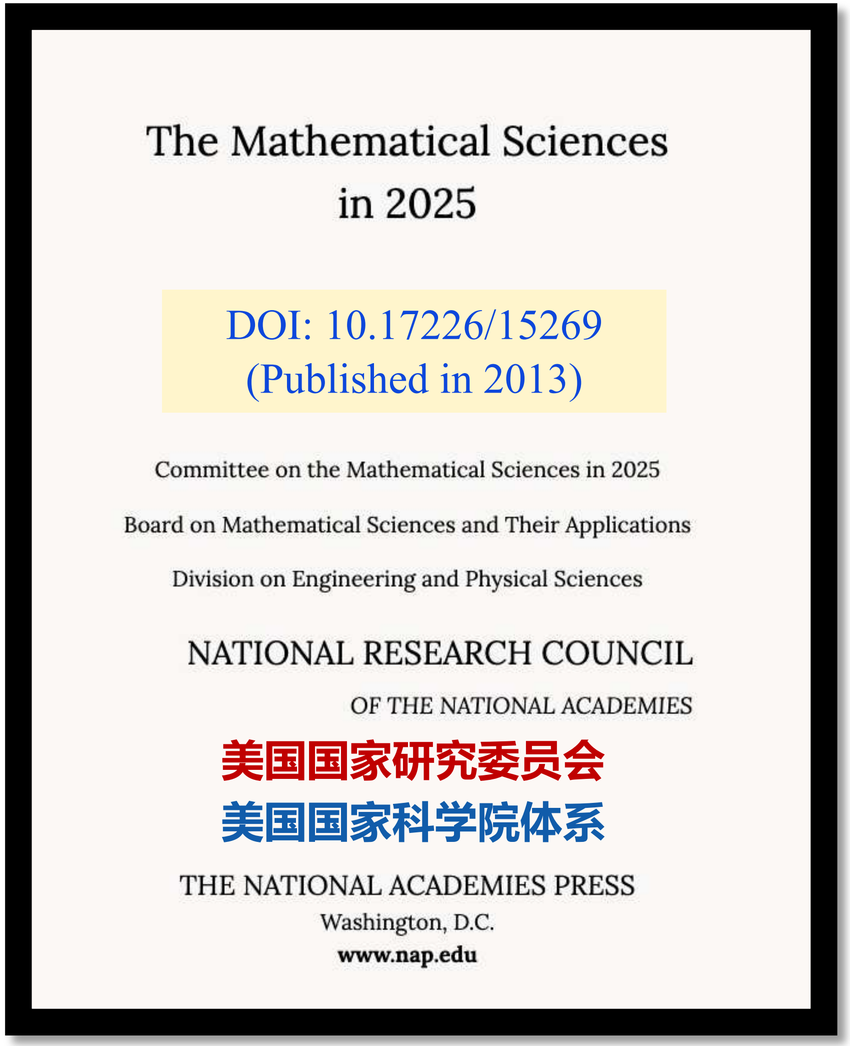
课程介绍

About this course

Chensong Zhang, AMSS

https://lsec.cc.ac.cn/~zhangcs

Mathematical Sciences



The Mathematical Sciences in 2025

DOI: 10.17226/15269
(Published in 2013)

Committee on the Mathematical Sciences in 2025

Board on Mathematical Sciences and Their Applications

Division on Engineering and Physical Sciences

NATIONAL RESEARCH COUNCIL

OF THE NATIONAL ACADEMIES

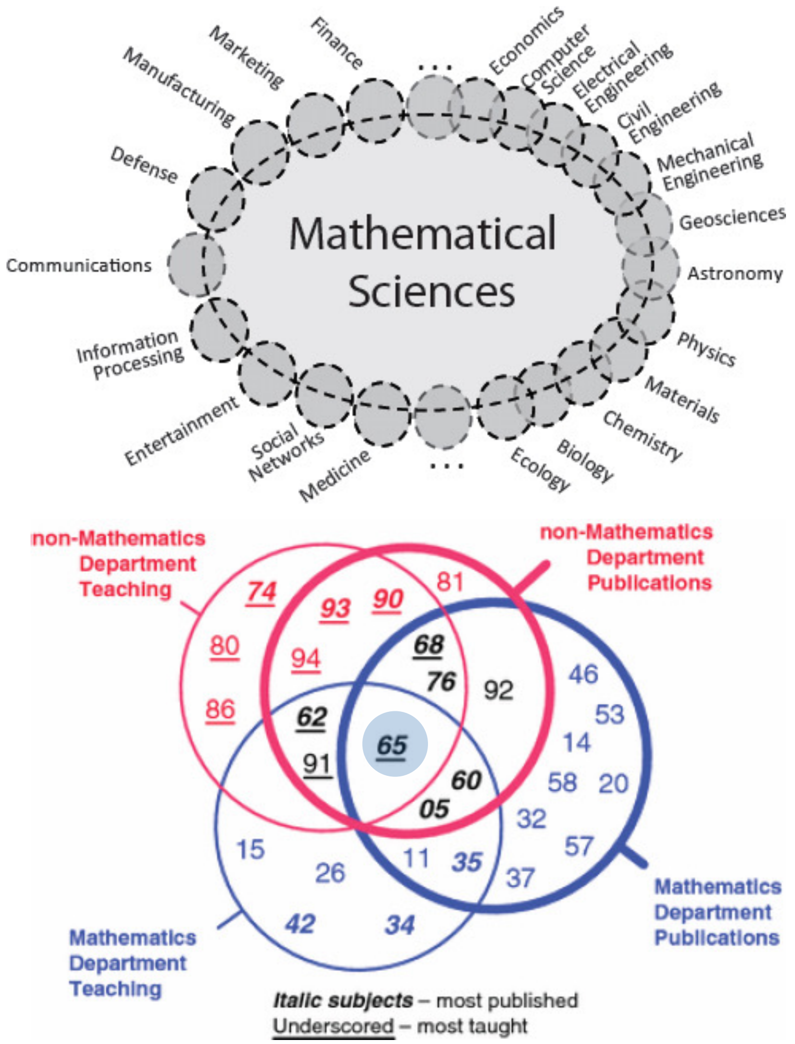
美国国家研究委员会

美国国家科学院体系

THE NATIONAL ACADEMIES PRESS

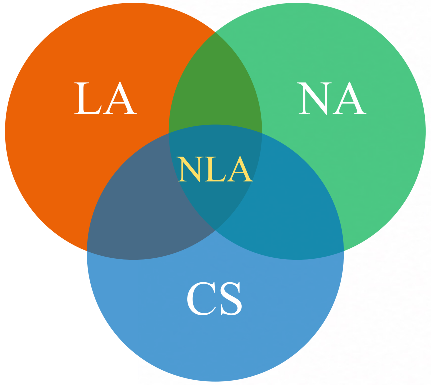
Washington, D.C.

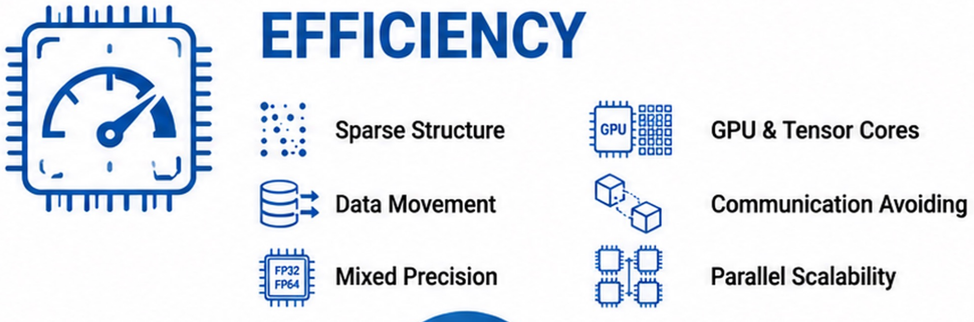
www.nap.edu

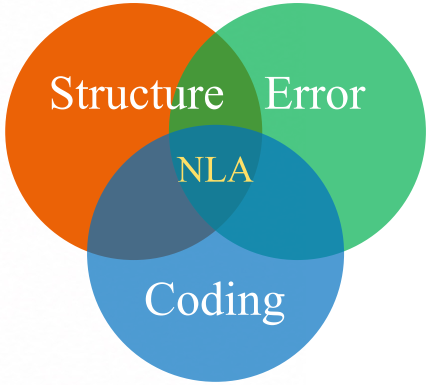


05 Combinatorics
11 Number theory
14 Algebraic geometry
15 Linear, multilinear algebra
20 Group theory
26 Real functions
32 Several complex variables
34 Ordinary differential equations
35 Partial differential equations
37 Dynamical systems
42 Fourier analysis
46 Functional analysis
53 Differential geometry
57 Manifolds, cell complexes
58 Global analysis
60 Probability theory
62 Statistics
65 Numerical analysis
68 Computer science
74 Mechanics of deformable solids
76 Fluid mechanics
80 Classical thermodynamics
81 Quantum theory
86 Geophysics
90 Operations research
91 Game theory, economics
92 Biology
93 Systems theory, control
94 Information and communications

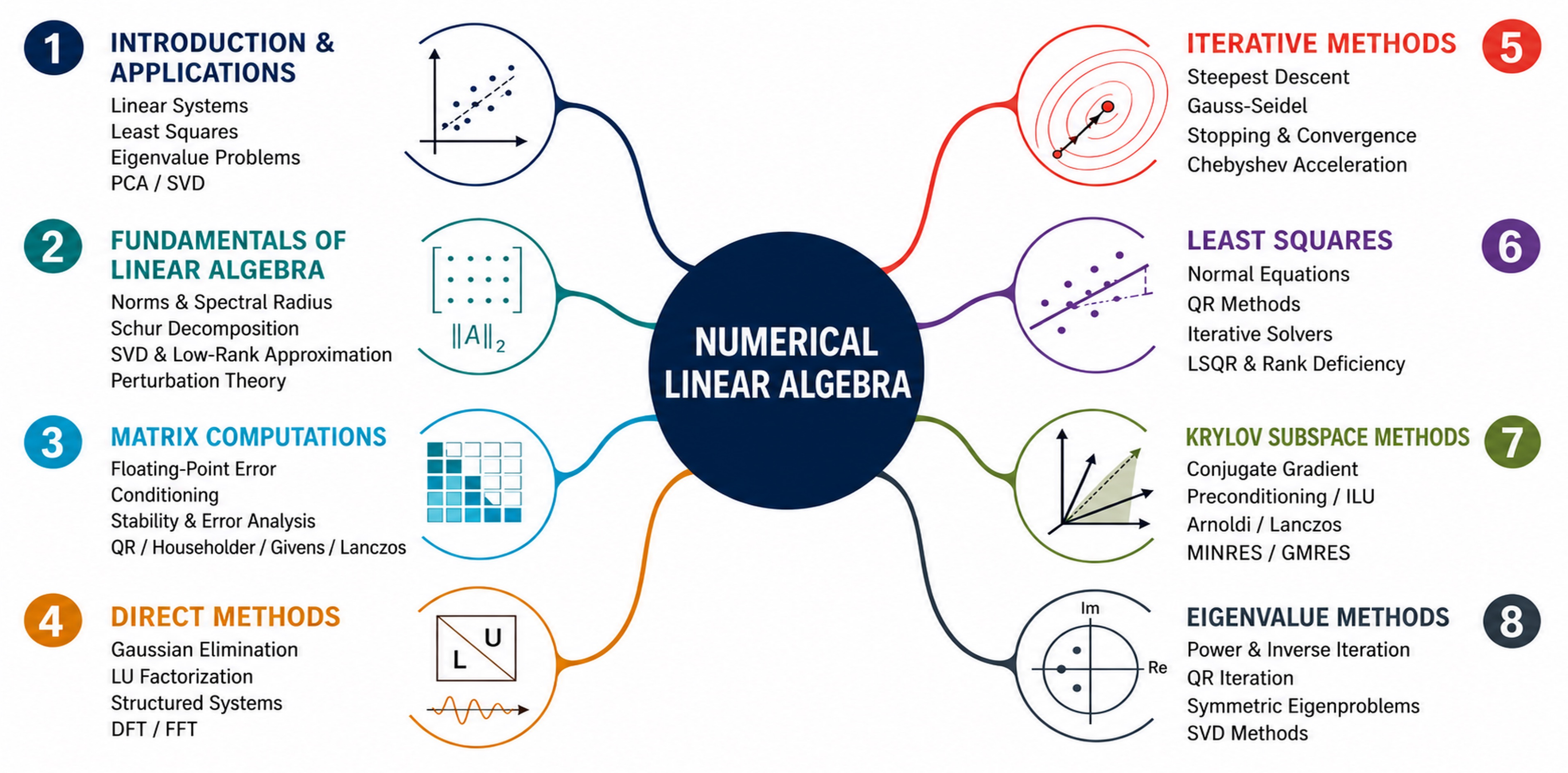
The Art of NLA







Mind Map for NLA



Education with AI

能力教育的四个境界

港科大（广州）熊辉校长｜《AI时代的职业图谱》演讲

01	02	03	04
博文强识	**触类旁通**	**一叶知秋**	**无中生有**
广泛积累知识与技能	跨领域迁移能力	由细节洞察规律	由问题驱动创造

AI时代的能力闭环

问 **提问能力**
触发 AI 输出

鉴 **鉴赏能力**
校正 AI 输出

把能力变成动作

发现问题 → 形成问题 → 分解问题

好的提问，打开输出；好的鉴赏，决定答案能否成为行动

AI is Powerful, Yet “I” Decide

思辨 (Critical Thinking)

涵盖判断能力与怀疑精神，保持独立思考、质疑和判断的能力

创新 (Innovation)

即创新精神，是从0到1的原创能力和跨领域整合的能力

品味 (Taste)

代表审美、价值判断和情感共鸣的能力，是人类独特的财富

协作 (Collaboration)

指人机合作能力，明确人类与AI的分工，让AI成为能力的放大器

重新定义“教学”：内容与方法

重构课程体系

- 强化人类独有能力（思辨、创新、品味）
- 普及AI通识教育，理解技术边界，提升人机协作能力

AI赋能实战

- 利用AI实现个性化学习路径规划
- 推行项目式学习，在实战中掌握人机协作

田刚“三个能力”（大湾区大学2026开学典礼）：提出问题的能力、独立思考的能力、终身学习的能力

Human Comprehension as a Gatekeeper

Stephen Wolfram (2024)

Creator of Mathematica and Wolfram Alpha

"Without such interpretive landmarks, we merely have a black-box computation, not assimilable human knowledge."

Source: StephenWolframWritings, "Can AI Solve Science?"

Terence Tao (2026)

Fields Medal Winner

"... a verified proof of a major result that no human understands well enough to explain... The bottleneck shifts from generating proofs to human understanding."

Source: arXiv:2608.08562

Springer-Nature AI Publishing Policy (2025)

Human accountability is non-transferable... Authors must be able to understand, verify and take responsibility for all content.

The Future of Knowledge

Linus Torvalds (2026)

Father of Linux and Git

"It is not acceptable to merge code that passes tests but cannot be understood... by human maintainers."

Source: Open Source Summit NA 2026

Uncle Bob Martin (2026)

Originator of Clean Code principles

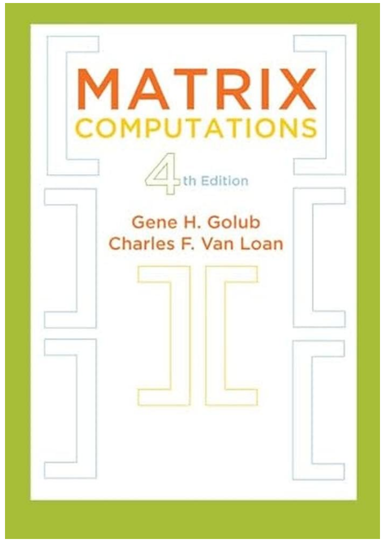
"We do not need to read every line of implementation code generated by AI agents... Humans own specifications, architecture, test definitions."

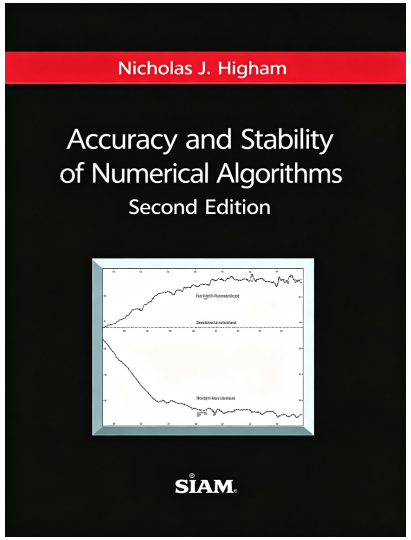
Source: Swarm-Forge Workflow

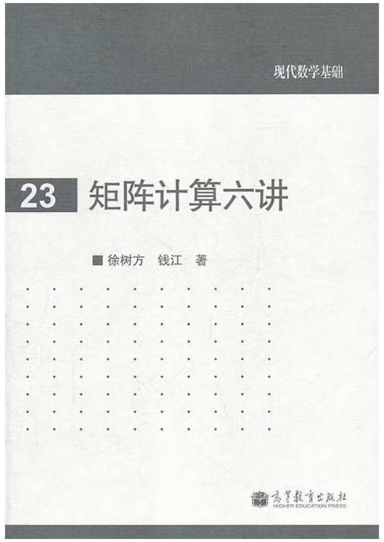
Takeaway Message: These perspectives are not mutually exclusive. The key takeaway is that human oversight remains paramount—whether at the level of deep understanding of complex proofs or defining robust tests that guide AI systems.

References

- Golub G H, Van Loan C F. **Matrix Computations**, 4^{th} Edition. JHU Press, 2013
- 徐树方，矩阵计算的理论与方法，北京大学出版社，1995
- Demmel J W. **Applied Numerical Linear Algebra**. SIAM, 1997
- Higham N J. **Accuracy and Stability of Numerical Algorithms**, 2^{nd} Edition. SIAM, 2002
- 徐树方、钱江，矩阵计算六讲，高等教育出版社, 2011







Grades

Exercises

算法、理论、推导

20%

Homework

实践、编程、应用

30%

Final

计算、证明

50%

Contact Me

Office hours: By appointment

zhangcs@lsec.cc.ac.cn

https://lsec.cc.ac.cn/~zhangcs

中国科学院大学

2026

数值代数

Numerical Algebra

THANKS

Chensong Zhang, AMSS

https://lsec.cc.ac.cn/~zhangcs

Release version 2026.089.14

中国科学院大学

2026

数值代数

Numerical Algebra

第一章、绪论与应用背景

Introduction and Applications

1

Chensong Zhang, AMSS

https://lsec.cc.ac.cn/~zhangcs

课程内容

第一章、绪论与应用背景

第二章、线性代数基础

第三章、矩阵计算基础

第四章、线性方程组的直接方法

第五章、线性方程组的迭代方法

第六章、最小二乘问题的数值方法

第七章、Krylov子空间方法

第八章、特征值问题的数值方法

本章主要内容

第一章、绪论与应用背景

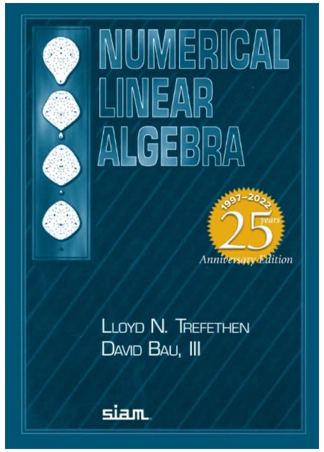
本章作为课程导论，介绍数值线性代数的核心问题及其在科学与工程中的实际背景。内容涵盖线性空间、线性变换与矩阵的基本概念，并通过典型应用，如线性回归、主成分分析、PageRank等，展示数值线性代数的重要性。同时，讨论数值实现中的一些关键挑战，包括浮点运算、稀疏矩阵处理、数据移动与并行效率等，为学生建立课程的整体框架。

基础知识

Basics of numerical linear algebra

/1.1

Numerical Linear Algebra



- Real-world problems are too complex for pen-and-paper solutions.
- L. N. Trefethen and D. Bau, III on **NLA** (1997):

> "If any other mathematical topic is as fundamental to the mathematical sciences as calculus and differential equations, it is numerical linear algebra."

- **Some applications** (NLA or Applied Linear Algebra):
  - **Scientific/Engineering Computing:** Simulates physical systems governed by ODEs/PDEs/IEs
  - **Principal Component Analysis:** Uses SVD for dimensionality reduction and feature extraction
  - **Image Compression:** Uses the Discrete Cosine Transform, like in JPEG
  - **Recommendation Systems:** Uses matrix factorization techniques (a form of low-rank approximation)
  - **PageRank:** Models the web as a graph and finds the principal eigenvector of the link matrix
  - **Deep Learning:** Training DNNs involves massive matrix multiplications and solving optimization problems

Classical Problems in NLA

Numerical linear algebra (NLA) is the essential computational engine powering modern science and engineering, enabling scientific discoveries like large-scale simulations, AI, quantum computing, and financial analytics, via matrix decompositions, linear system solvers, eigenvalue solvers, etc.

- **Prob 1.** Linear System solving (**LS**) problems

$$\text{Find } \boldsymbol{x} \in \mathbb{C}^n \ : \ A\boldsymbol{x} = \boldsymbol{b}, \ \text{ where } A \in \mathbb{C}^{n\times n}, \boldsymbol{b} \in \mathbb{C}^n$$

- **Prob 2.** Least SQuares (**LSQ**) problems

$$\arg\min_{\boldsymbol{x}\in\mathbb{C}^n} \ \|A\boldsymbol{x} - \boldsymbol{b}\|_2^2, \ \text{ where } A \in \mathbb{C}^{m\times n}, \boldsymbol{b} \in \mathbb{C}^m$$

- **Prob 3.** EigenValue problems (**EVP**)

$$\text{Find } \lambda \in \mathbb{C}, \ \boldsymbol{0} \neq \boldsymbol{x} \in \mathbb{C}^n \ \text{ s.t. } \ A\boldsymbol{x} = \lambda\boldsymbol{x}, \ \text{ where } A \in \mathbb{C}^{n\times n}$$

Other Problems in NLA

- **Prob 4. Singular Value Problems**

$$\text{Find singular triplets } (\sigma_i, \boldsymbol{u}_i, \boldsymbol{v}_i) \ : \ A\boldsymbol{v}_i = \sigma_i \boldsymbol{u}_i, \ A^H \boldsymbol{u}_i = \sigma_i \boldsymbol{v}_i$$

such that $\sigma_i \geq 0, \ \|\boldsymbol{u}_i\| = \|\boldsymbol{v}_i\| = 1$

- **Prob 5. Low-Rank Approximation Problems**

$$\min_{\mathrm{rank}(X) \leq k} \|A - X\|_F$$

- **Prob 6. Matrix Function Action Problems**

$$\text{Compute } \ \boldsymbol{y} = f(A)\boldsymbol{b}, \quad \text{e.g. } \ f(t) = e^t, \ t^\alpha, \ \log(t), \ldots$$

AI and data science: matrix completion, sparse representation, spectral graph learning, kernel methods, …

Core building blocks: BLAS, LU, Cholesky, QR, SVD, Schur, and other related matrix factorizations, …

典型应用问题

Some practical applications

/1.2

Linear Regression

- Regression problem:

  To predict a **continuous model** of a target variable based on one or more input features

  $$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$$

  where $\boldsymbol{x}_i \in \mathbb{R}^d, \; y_i \in \mathbb{R}, \; i = 1 : n.$

- Linear regression model:

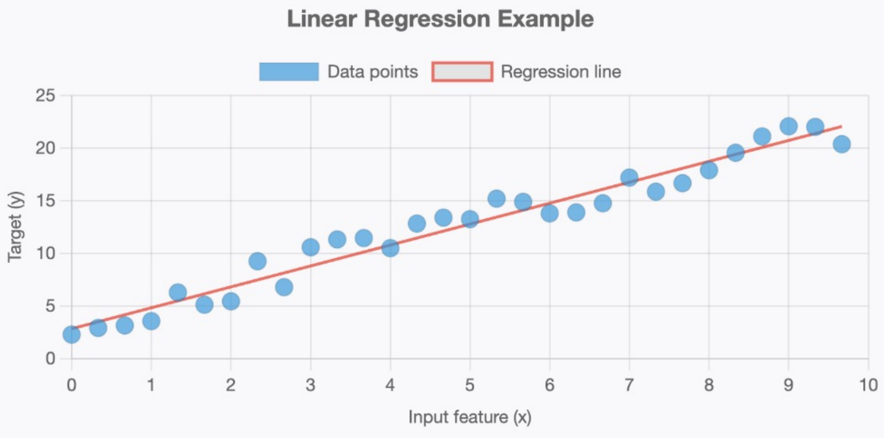
  $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d + \varepsilon$$

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{d+1}} \|\boldsymbol{y} - X\boldsymbol{\beta}\|^2 \quad \longrightarrow \quad X^T X \boldsymbol{\beta} = X^T \boldsymbol{y}$$

$$X := \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix}$$

Provided that X has full column rank

Model: y = 2x + 3 + noise



Regression line: $\hat{y} = 2.02x + 2.89$ ($R^2 = 0.92$)

Practical Applications:

- Predicting house prices based on size, location, and amenities
- Forecasting temperature based on historical weather data
- Estimating blood pressure based on age, weight, and lifestyle
- Predicting stock prices based on market indicators

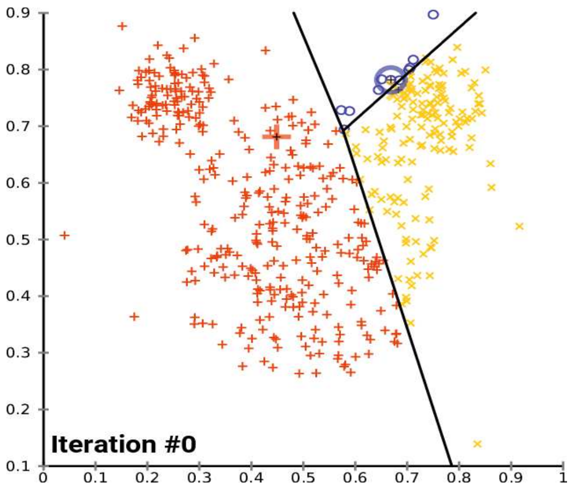
Clustering and Eigen Decomposition

- Clustering:

  Partition a set of data points into groups called clusters such that:

  - Points in the same cluster are **similar**
  - Points from different clusters are **dissimilar**

- k-Means clustering (MacQueen, J. 1967):



Limitations of k-means:

✗ Assumes clusters are convex and isotropic

✗ Sensitive to initialization

✗ Performs poorly with nonlinearly separable data

- Spectral Clustering:

1. Construct Similarity Graph:

Create affinity matrix W where $W_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$

2. Compute Graph Laplacian:

$$L = D - W$$

where D is the diagonal degree matrix: $D_{ii} = \sum_j W_{ij}$

3. Normalized Laplacian:

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

4. Eigenvalue Decomposition:

Compute the first k eigenvectors $v_1, v_2, \ldots, v_k$ of L_{sym}

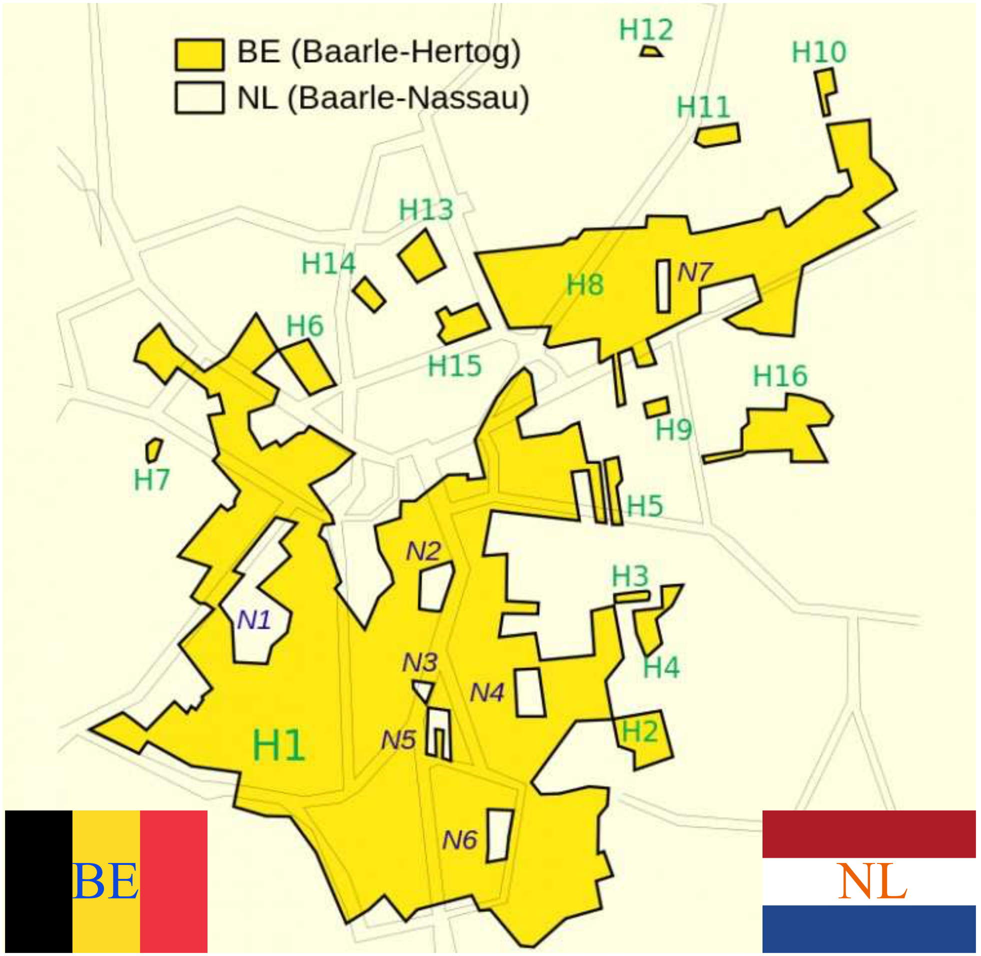
5. New Representation:

Form matrix $V \in \mathbb{R}^{n \times k}$ from the eigenvectors

6. Apply k-means:

Apply k-means to the rows of V to obtain clusters

A More Involved Example



Baarle-Hertog/Nassau:

The World's Most Confusing Border

The twin municipalities of Baarle-Hertog (Belgium) and Baarle-Nassau (Netherlands) together form what is widely considered the most complex international border in the world. Within and around the town of Baarle, there are 30 separate enclaves — 22 Belgian parcels within the Netherlands and 8 Dutch parcels within those Belgian enclaves, creating enclaves within enclaves that defy conventional cartography.

Q: How to design a method to label the BE / NL parts?

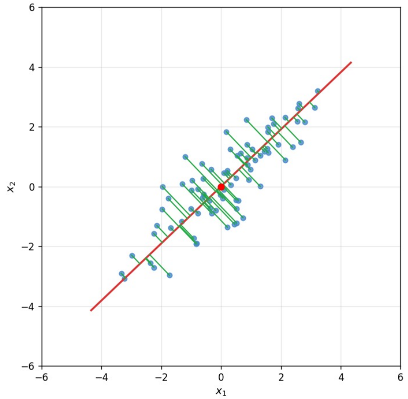
https://blogs.transparent.com/dutch/baarle-one-of-the-strangest-municipalities-on-earth/

Principal Component Analysis

- **Principal Component Analysis (PCA):**

Given centered data $X \in \mathbb{R}^{n \times p}$ (n samples, p features), PCA finds orthogonal directions that maximize variance, i.e.

$$V_k = \underset{V^T V = I_k}{\arg\max} \|XV\|_F^2$$



$$\boldsymbol{v}_1 = \underset{\|\boldsymbol{v}\|=1}{\arg\max} \sum_{i=1}^{n} (\boldsymbol{v}^T \boldsymbol{x}_i)^2$$

$p = 2, n = 80$

- Maximum projected variance

$\Longleftrightarrow$ Minimum orthogonal reconstruction error

$$\underset{V^T V = I_k}{\arg\max} \|XV\|_F^2 = \underset{V^T V = I_k}{\arg\min} \left\|X - XVV^T\right\|_F^2$$

EVD-Based Method

1. Center data: $X_c = X - \bar{X}$
2. Compute covariance matrix: $C = \frac{1}{n-1} X_c^T X_c$
3. Eigen decomposition: $C = W \Lambda W^T$
4. Select top-k eigenvectors (columns of W)
5. Project: $T = X_c W_k$

SVD-Based Method

1. Center data: $X_c = X - \bar{X}$
2. Compute SVD: $X_c = U \Sigma V^T$
3. Principal components: Columns of V
4. Scores: $T = U\Sigma$ or $T = X_c V$

<u>Practical Applications:</u>

- Data visualization (reducing to 2D/3D)
- Noise reduction
- Feature extraction
- Speeding up machine learning algorithms

PageRank and Eigenvectors

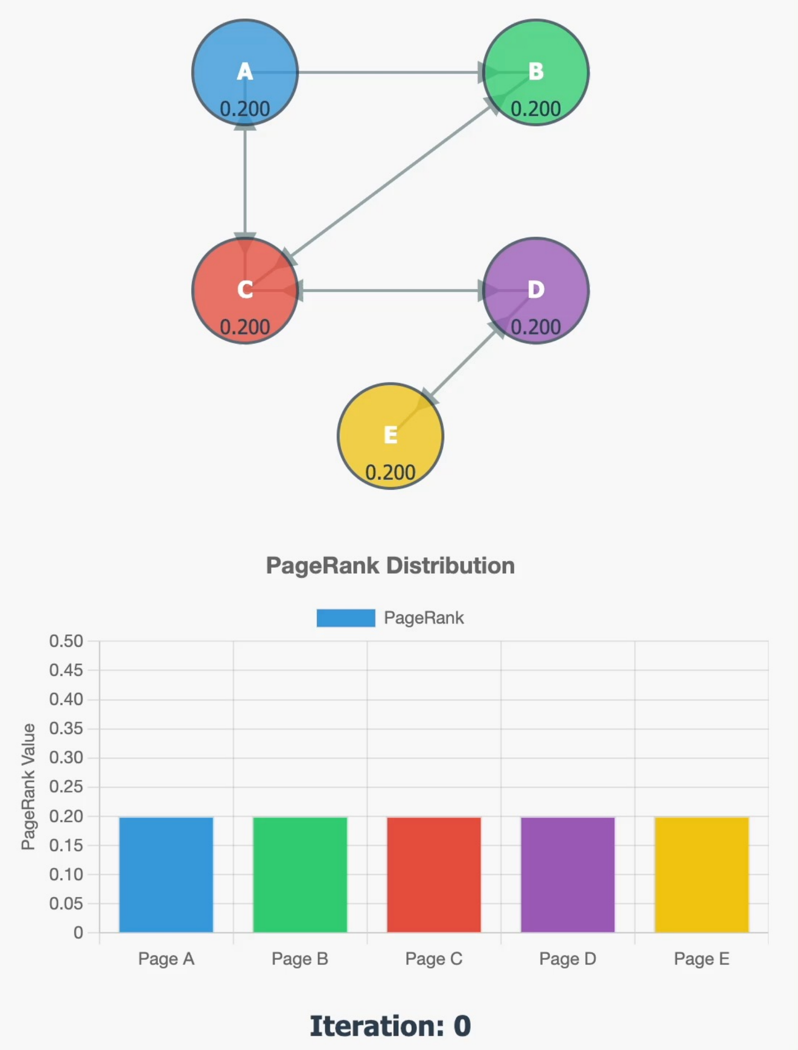
- PageRank:

PageRank is a link analysis algorithm developed by Google founders (1998). A random surfer follows an outgoing link with probability d, and jumps uniformly to any page with probability $1-d$.

$$PR(A) = \frac{1-d}{n} + d \sum_{B_i \to A} \frac{PR(B_i)}{C(B_i)}$$

Q: How to solve it?

- $PR(A)$ is the PageRank of page A
- $C(B_i)$ is the number of outbound links from B_i



L. Page & S. Brin, 1998: Find the principal eigenpair of the so-called Google matrix (need to handle dangling node j by setting $M_{:,j} = 1/n$):

$$G := d\,M + \frac{1-d}{n}\mathbf{1}\mathbf{1}^T, \quad M_{ij} := \begin{cases} 1/C(j), & j \to i \\ 0, & \text{otherwise} \end{cases}$$

The spectral radius of G is 1 and its eigenvector (PageRank vector) is positive.

Landscape of Parallel Computing

The Landscape of Parallel Computing Research: A View from Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Krste Asanovic
Ras Bodik
Bryan Christopher Catanzaro
Joseph James Gebis
Parry Husbands
Kurt Keutzer
David A. Patterson
William Lester Plishker
John Shalf
Samuel Webb Williams
Katherine A. Yelick

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-183
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html

December 18, 2006

3.2.1 Machine Learning

One of the most promising areas for the future of computing is the use of statistical machine learning to make sense from the vast amounts of data now available due to faster computers, larger disks, and the use of the Internet to connect them all together.

Michael Jordan and Dan Klein, our local experts in machine learning, found two dwarfs that should be added to support machine learning:

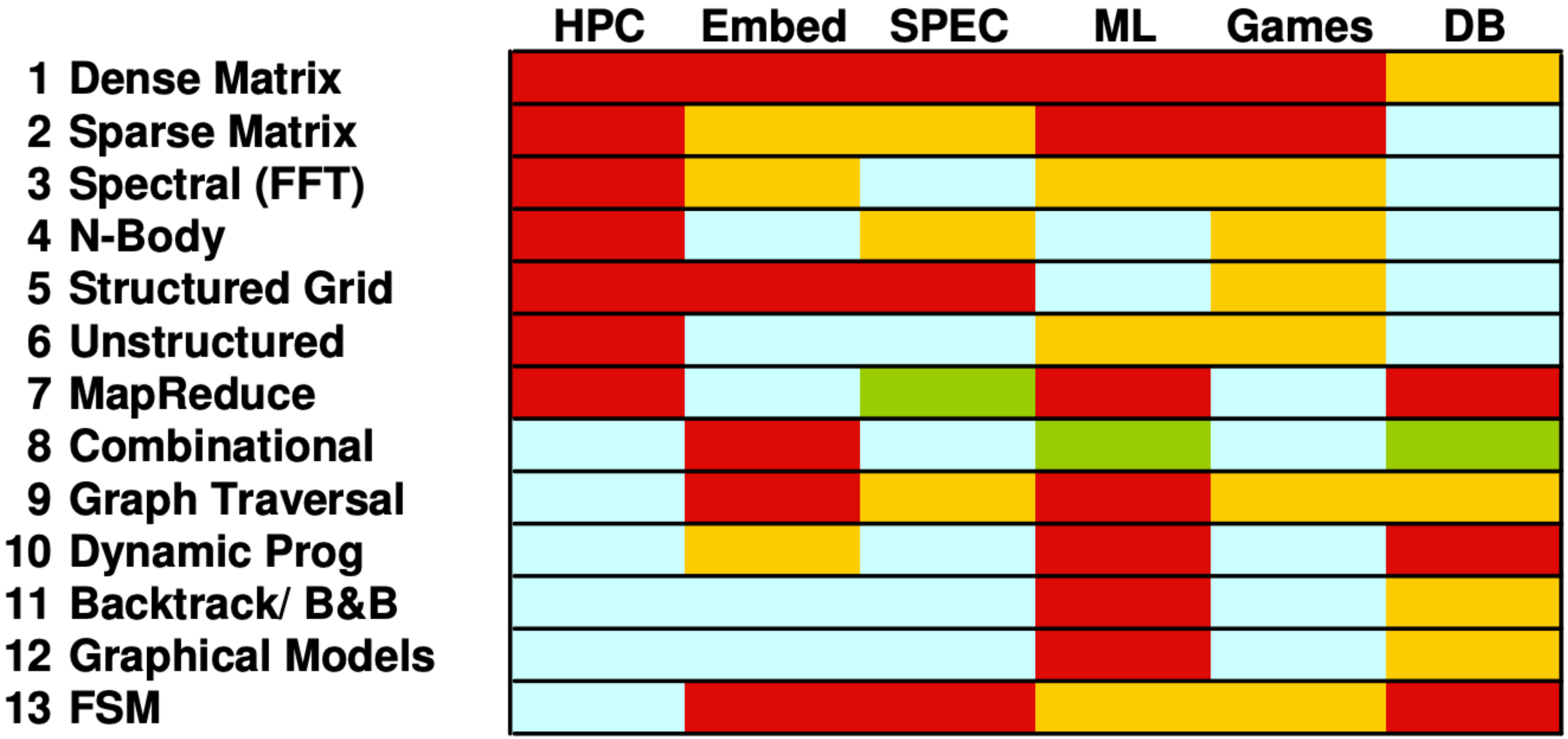
- **Dynamic programming** is an algorithmic technique that computes solutions by solving simpler overlapping subproblems. It is particularly applicable for optimization problems where the optimal result for a problem is built up from the optimal result for the subproblems.
- **Backtrack and Branch-and-Bound**: These involve solving various search and global optimization problems for intractably large spaces. Some implicit method is required in order to rule out regions of the search space that contain no interesting solutions. Branch and bound algorithms work by the divide and conquer principle: the search space is subdivided into smaller subregions ("branching"), and bounds are found on all the solutions contained in each subregion under consideration.

"To maximize programmer productivity, future programming models must be more human-centric than the conventional focus on hardware or applications."

Thirteen Dwarfs in Parallel Computing

Dwarf Name	Description	Examples	Performance Limitations (Current Understanding)
Dense Linear Algebra	Operations on dense matrices/vectors with regular memory access patterns	BLAS, ScaLAPACK	Computationally limited; modern GPUs and TPUs excel here
Sparse Linear Algebra	Operations on matrices with many zero values using compressed storage formats	SpMV, OSKI, SuperLU	50% computation, 50% memory bandwidth; hybrid approaches work best
Spectral Methods	Computations in frequency domain rather than time or spatial domains	FFT, Cooley-Tukey algorithm	Memory latency limited; can be optimized with cache-oblivious algorithms
N-Body Methods	Calculations based on interactions between discrete points	Barnes-Hut, Fast Multipole	Computationally limited; use spatial partitioning (octrees, k-d trees)
Structured Grids	Regular grid computations where points are updated together	Cactus, Lattice-Boltzmann	Memory bandwidth limited; excellent scaling with proper blocking
Unstructured Grids	Irregular grid where data locations and connectivity must be explicit	ABAQUS, FIDAP	Memory latency limited; partitioning and graph-based approaches
MapReduce	Parallel execution on independent data sets with combined results	Monte Carlo, Google MapReduce	Problem dependent; communication overhead becomes dominant at scale
Combinational Logic	Functions implemented with logical operations and stored state	Encryption, CRC computation	CRC problems bandwidth limited; excellent for SIMD/ASIC acceleration
Graph Traversal	Visiting nodes in a graph by following edges	Breadth-first search (BFS), DFS	Memory latency limited; significant gains from prefetching and locality
Dynamic Programming	Solving problems by combining solutions to overlapping subproblems	Optimization algorithms	Memory latency limited; can be parallelized with wavefront approaches, but suffers from synchronization overhead
Backtrack and Branch-and-Bound	Finding optimal solutions by recursive subdivision and pruning	Search, global optimization	Memory latency limited with synchronization overhead; irregular workloads and load imbalance issues
Graphical Models	Graphs representing random variables as nodes and dependencies as edges	Bayesian networks, Hidden Markov Models	Memory bandwidth limited with communication overhead
Finite State Machine	Systems defined by states, transitions based on inputs and current state	Parsing algorithms, protocol implementations	"Nothing helps!"; remains difficult to parallelize; modern approaches use speculative execution and state partitioning

Dwarf Popularity (Red Hot → Blue Cool)



Source: David Patterson, https://web.stanford.edu/class/ee380/Abstracts/070131-BerkeleyView1.7.pdf

Foundation & Future of Scientific Computing

- Numerical Linear Algebra (NLA) is the cornerstone of scientific computing, more generally computational science: **numerical stability** and **fundamental understanding** matter more than **computational sophistication** alone.

- Today's Landscape:
  - Scientific computing: NLA kernels dominate runtime in climate modeling, quantum chemistry, and various other engineering simulations;
  - AI/ML backbone: Matrix operations power neural networks, transformers, optimization algorithms, etc.

- Pressing Imperatives:
  - Stability at scale: As systems become larger, numerical error accumulation requires new analysis;
  - Hardware-aware algorithms: NLA must evolve with novel architectures (quantum, neuromorphic);
  - Cross-domain integration: Bridging SC with AI demands ➔ unified numerical frameworks.