Evolution of MATLAB

Cleve Moler
MathWorks

ATPESC 2017
St. Charles, IL
August 1, 2017

Origins of MATLAB

1947 – 1975
Before MATLAB

1976 – 1984
"Classic" MATLAB

1984 – 2016
MathWorks

Mentors

John Todd
George Forsythe
J. H. Wilkinson

1947

Alan Turing

National Physical Laboratory

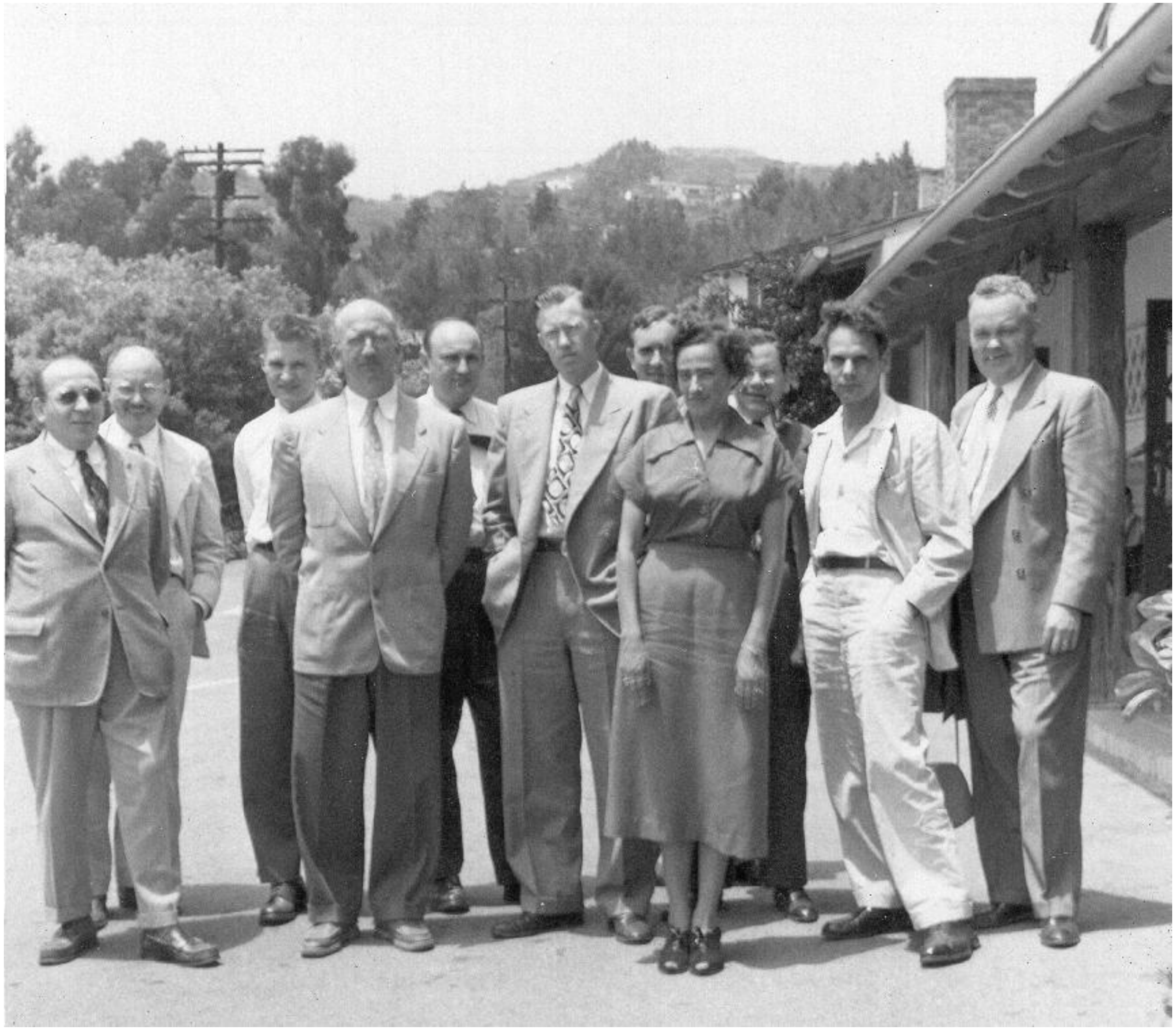
ACE

1951
J. H. Wilkinson
Pilot Ace

1952
SWAC
UCLA

Standards Western Automatic Computer

1952
INA
UCLA
George Forsythe
John Todd
Olga Taussky-Todd

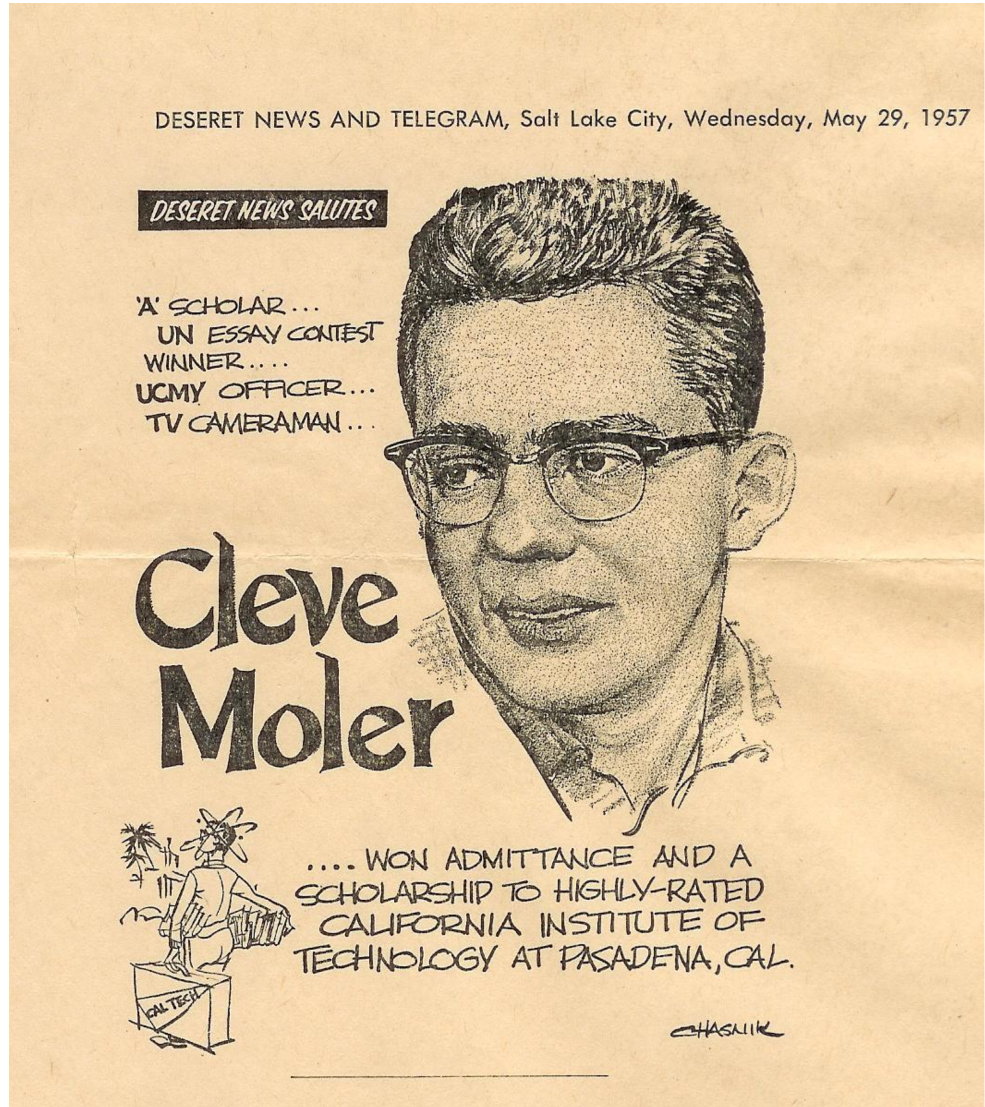


Late 1950's Caltech Math

Back Row, left to right: S. A. Andrea, F. B. Fuller, M. P. Drazin, C. R. B. Wright. **2nd Row: G. D. Chakeman, C. R. DePrima, T. M. Apostol, C. H.** Wilcox, R. P. Dilworth, M. Ward, J. Todd. **Front Row:** J. W. Macki, E. Tully, **Jr., A. Erdelyi, H. F. Bohnenblust, A. C. Zaanen, M. Hall, Jr., O. T. Todd,** W. A. J. Luxemburg.

MATHEMATICS

1957
Utah kid goes to Caltech


DESERET NEWS AND TELEGRAM, Salt Lake City, Wednesday, May 29, 1957

DESERET NEWS SALUTES

'A' SCHOLAR . . .
UN ESSAY CONTEST WINNER . . . .
UCMY OFFICER . . .
TV CAMERAMAN . . .

Cleve Moler

. . . . WON ADMITTANCE AND A SCHOLARSHIP TO HIGHLY-RATED CALIFORNIA INSTITUTE OF TECHNOLOGY AT PASADENA, CAL.

CHASNIK

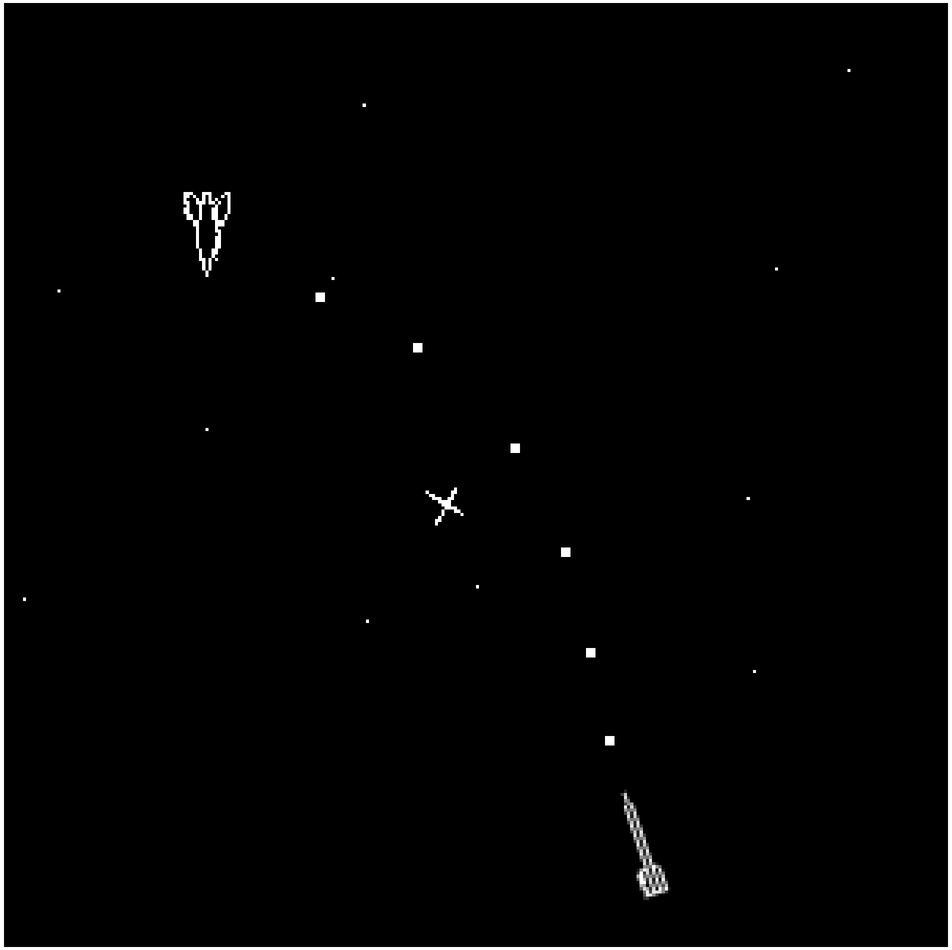
1959
Caltech
Burroughs 205

Machine language
Paper tape
Hilbert matrices

George Forsythe
Stanford
Math Professor
1964 Founds Computer
Science Department

1963
DEC PDP-1

1963
Space War



1964
Gatlinburg, TN

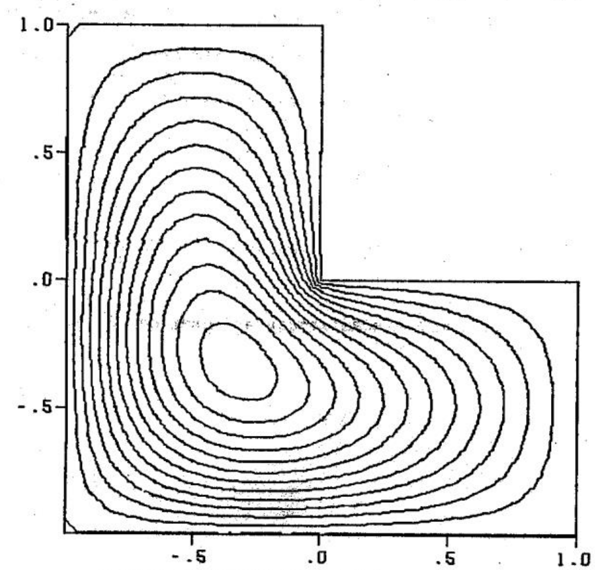
Wilkinson
Givens
Forsythe
Householder
Henrici
Bauer

1965
Stanford thesis
L-shaped membrane

L-Shaped Membrane

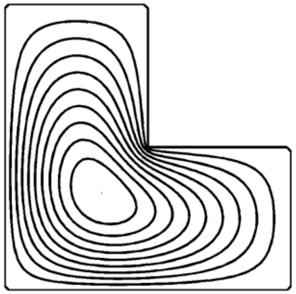
	h	Calculated λ_h^1	Estimated number of incorrect figures
	1/10	9.68829 144629	0.0
	1/20	9.66696 983477	1.2
	1/30	9.65743 367723	1.9
	1/40	9.65249 357713	2.4
	1/50	9.64954 711152	2.8
(8.14)	1/60	9.64761 706018	3.1
	1/70	9.64626 725003	3.4
	1/80	9.64527 693133	3.6
	1/90	9.64452 301324	3.8
	1/100	9.64393 241120	4.0

(8.15)

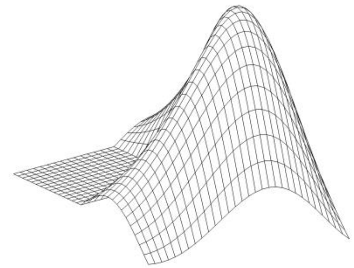


BASIC EIGENFUNCTION OF L. · N=20. · ·

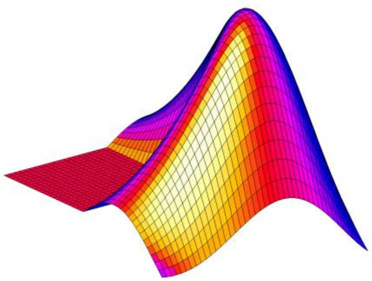
124



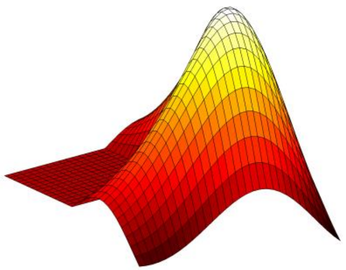
1984



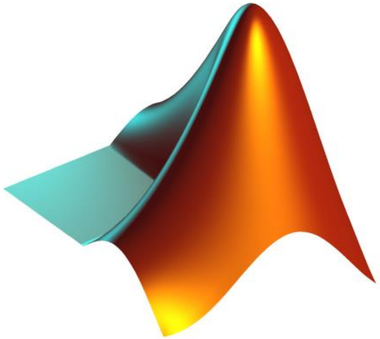
1987



1992

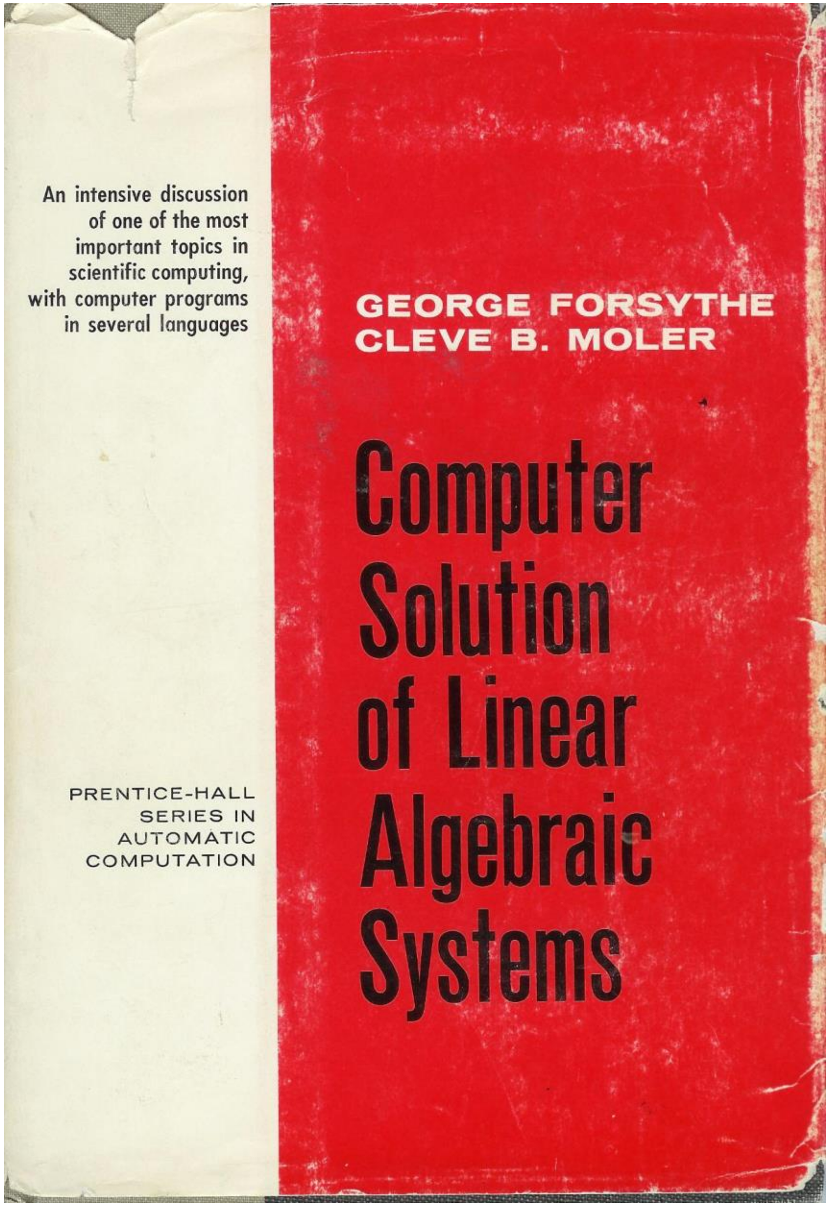


1994

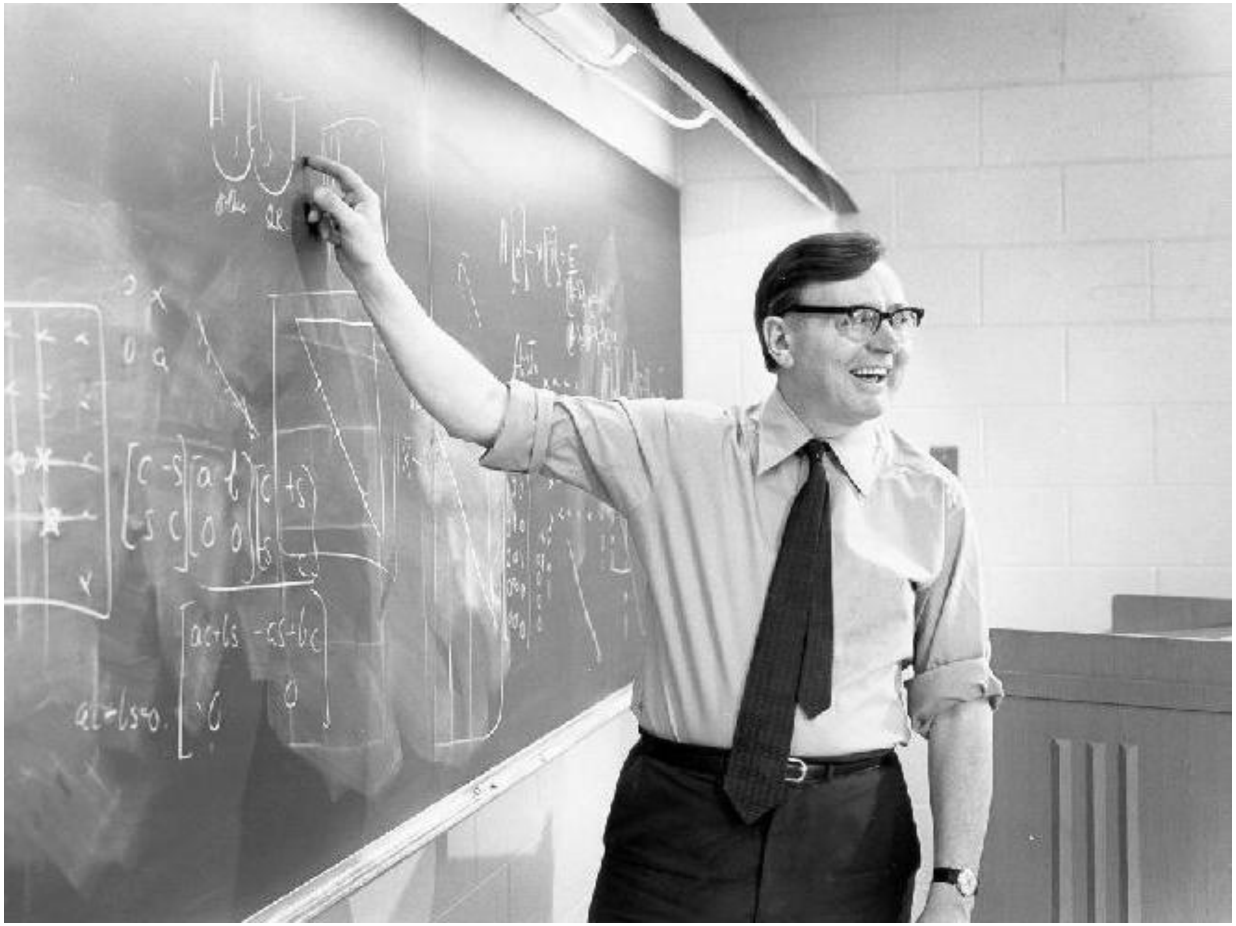


1997

1967
Forsythe & Moler



1960's-1970's
Wilkinson
Univ. Michigan
Argonne Nat'l Lab



MathWorks

1971
Wilkinson & Reinsch

Handbook for Automatic Computation

Edited by

F. L. Bauer · A. S. Householder · F. W. J. Olver
H. Rutishauser † · K. Samelson · E. Stiefel

Volume II

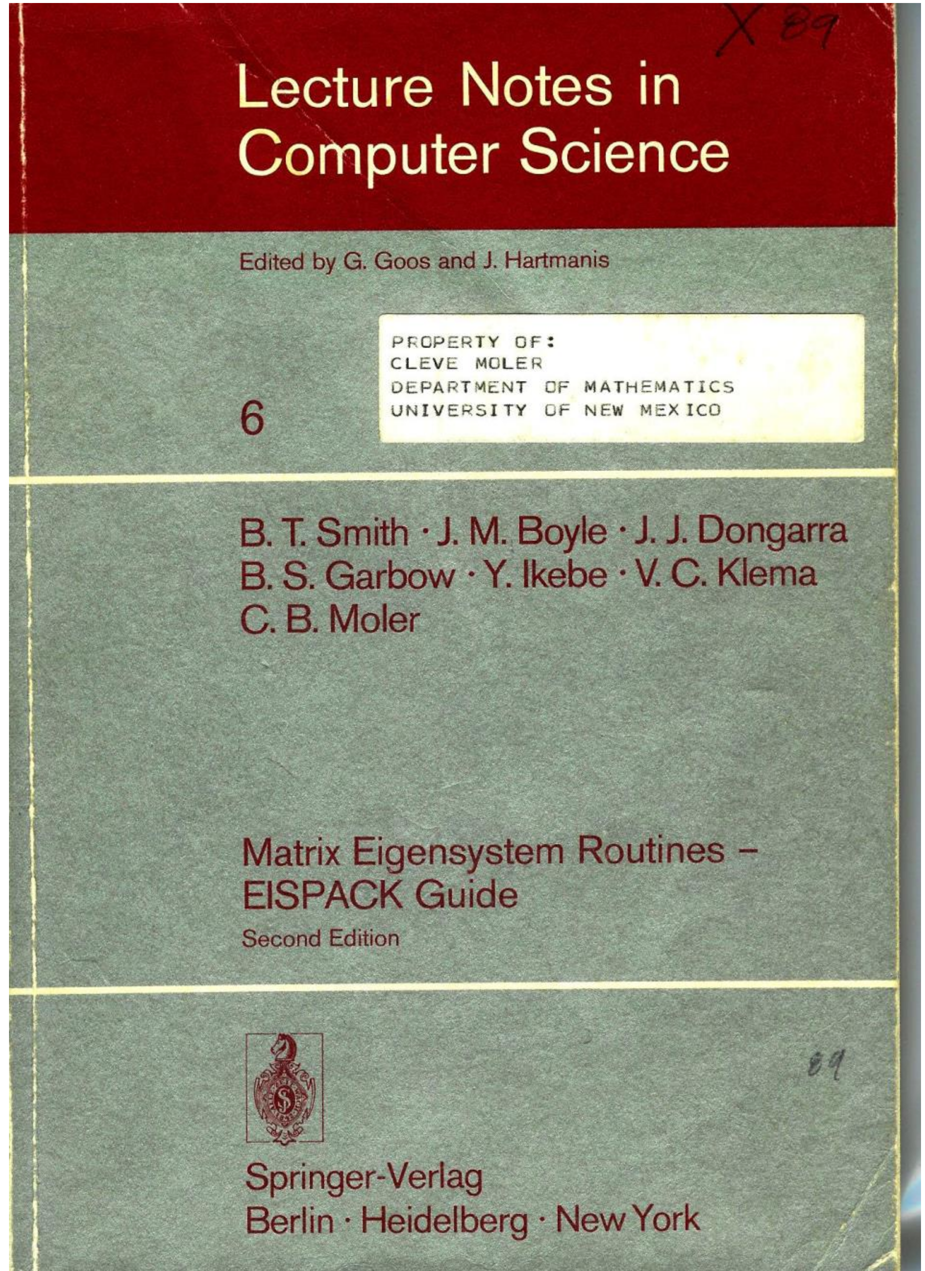
J. H. Wilkinson · C. Reinsch

Linear Algebra

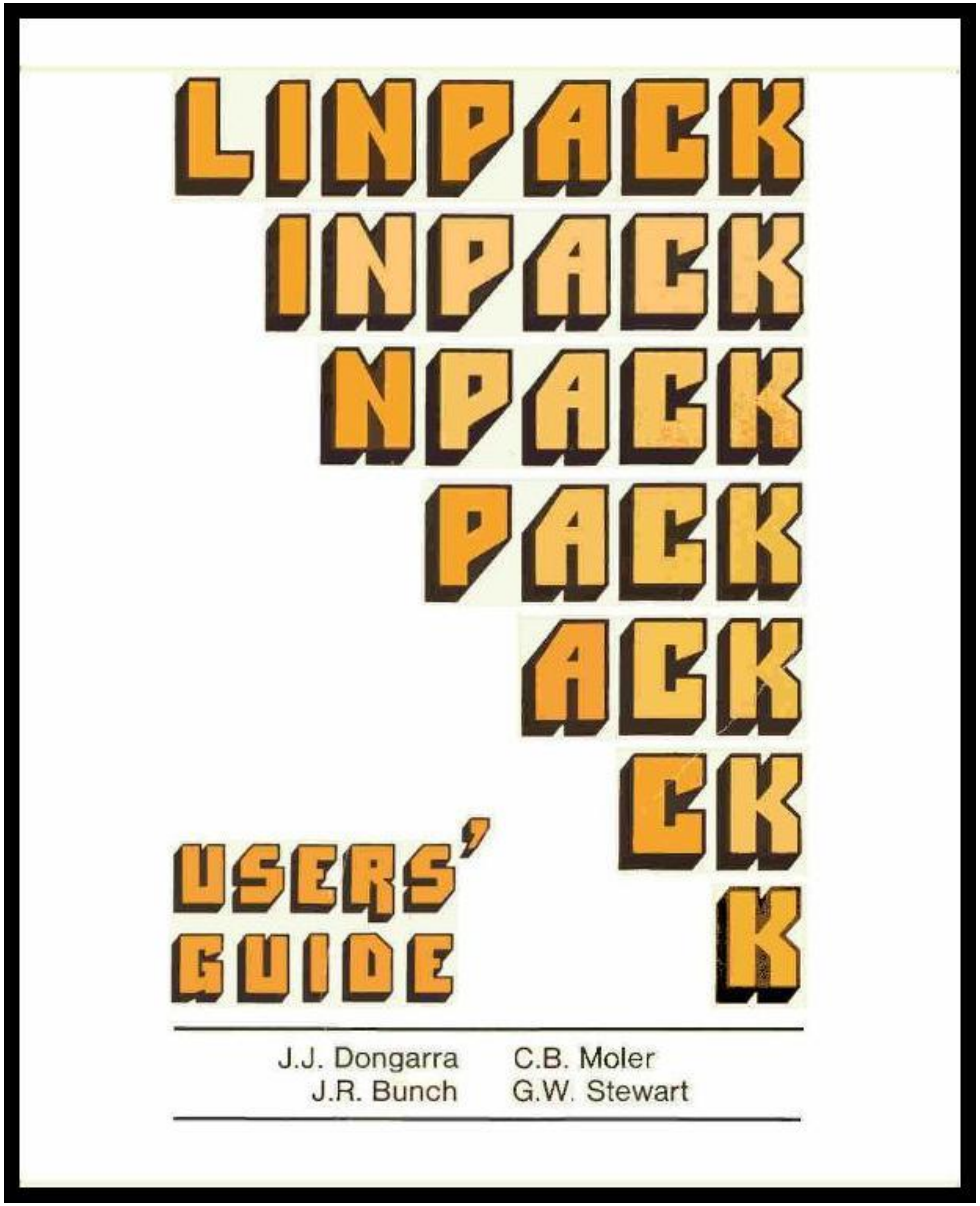
Chief editor
F. L. Bauer

Springer-Verlag Berlin Heidelberg New York 1971

1976
EISPACK



1979
LINPACK



1979
Jack Dongarra
Cleve Moler
Pete Stewart
Jim Bunch

2011
33 Years
Later

Appendix B First LINPACK Benchmark

$\frac{2}{3}n^3 + 2n^2$ ops / time

UNIT = 10**6 TIME/(1/3 100**3 + 100**2)

Facility		TIME N=100 secs.	UNIT micro-secs.	Computer	Type	Compiler
NCAR	14.0	.049	0.14	CRAY-1	S	CFT, Assembly BLAS
LASL	4.64	.148	0.43	CDC 7600	S	FTN, Assembly BLAS
NCAR	3.58	.192	0.56	CRAY-1	S	CFT
LASL	3.27	.210	0.61	CDC 7600	S	FTN
Argonne	2.31	.297	0.86	IBM 370/195	D	H
NCAR	1.91	.359	1.05	CDC 7600	S	Local
Argonne	1.77	.388	1.33	IBM 3033	D	H
NASA Langley	1.40	.489	1.42	CDC Cyber 175	S	FTN
U. Ill. Urbana	1.36	.506	1.47	CDC Cyber 175	S	Ext. 4.6
LLL	1.24	.554	1.61	CDC 7600	S	CHAT, No optimize
SLAC	1.19	.579	1.69	IBM 370/168	D	H Ext., Fast mult.
Michigan	1.09	.631	1.84	Amdahl 470/V6	D	H
Toronto	.772	.890	2.59	IBM 370/165	D	H Ext., Fast mult.
Northwestern	.477	1.44	4.20	CDC 6600	S	FTN
Texas	.356	1.93*	5.63	CDC 6600	S	RUN
China Lake	.352	1.95*	5.69	Univac 1110	S	V
Yale	.265	2.59	7.53	DEC KL-20	S	F20
Bell Labs	.199	3.46	10.1	Honeywell 6080	S	Y
Wisconsin	.197	3.49	10.1	Univac 1110	S	V
Iowa State	.194	3.54	10.2	Itel AS/5 mod3	D	H
U. Ill. Chicago	[illegible]	4.10	11.9	IBM 370/158	D	G1
Purdue	[illegible]	5.69	16.6	CDC 6500	S	FUN
U. C. San Diego	[illegible]	13.1	38.2	Burroughs 6700	S	H
Yale	[illegible]	17.1*	49.9	DEC KA-10	S	F40

* TIME(100) = (100/75)**3 SGEFA(75) + (100/75)**2 SGESL(75)

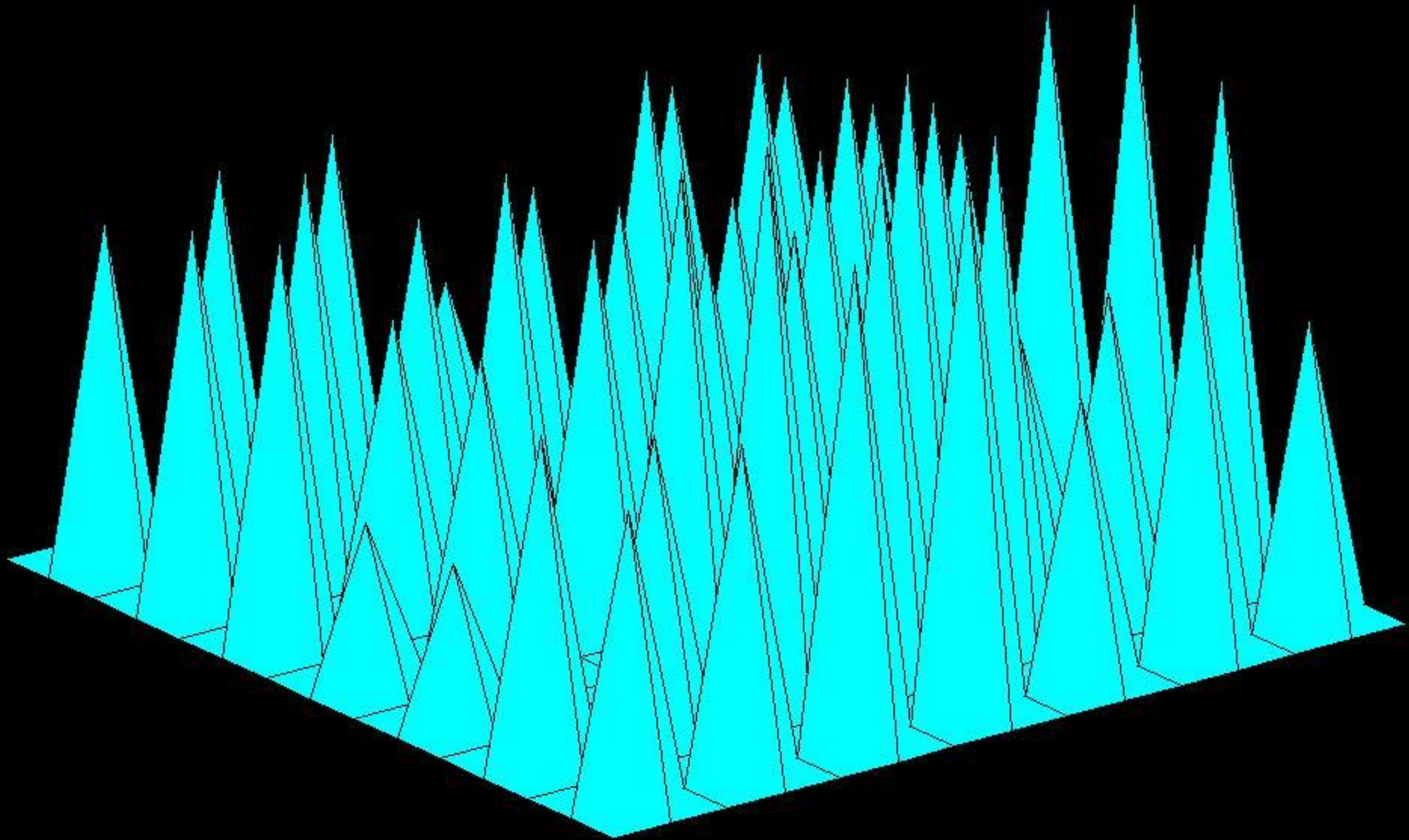
MathWorks®

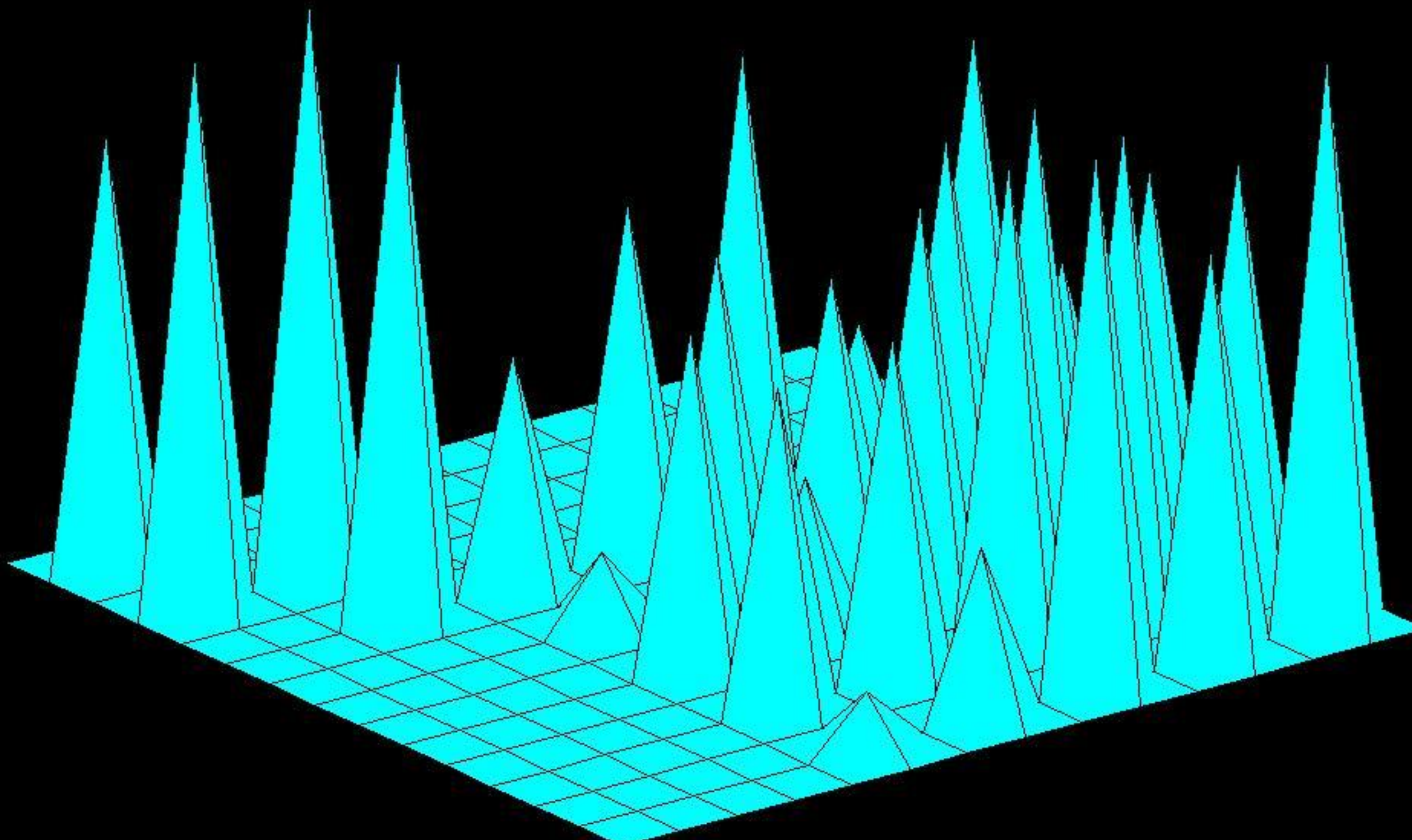
TOP 500
The List.

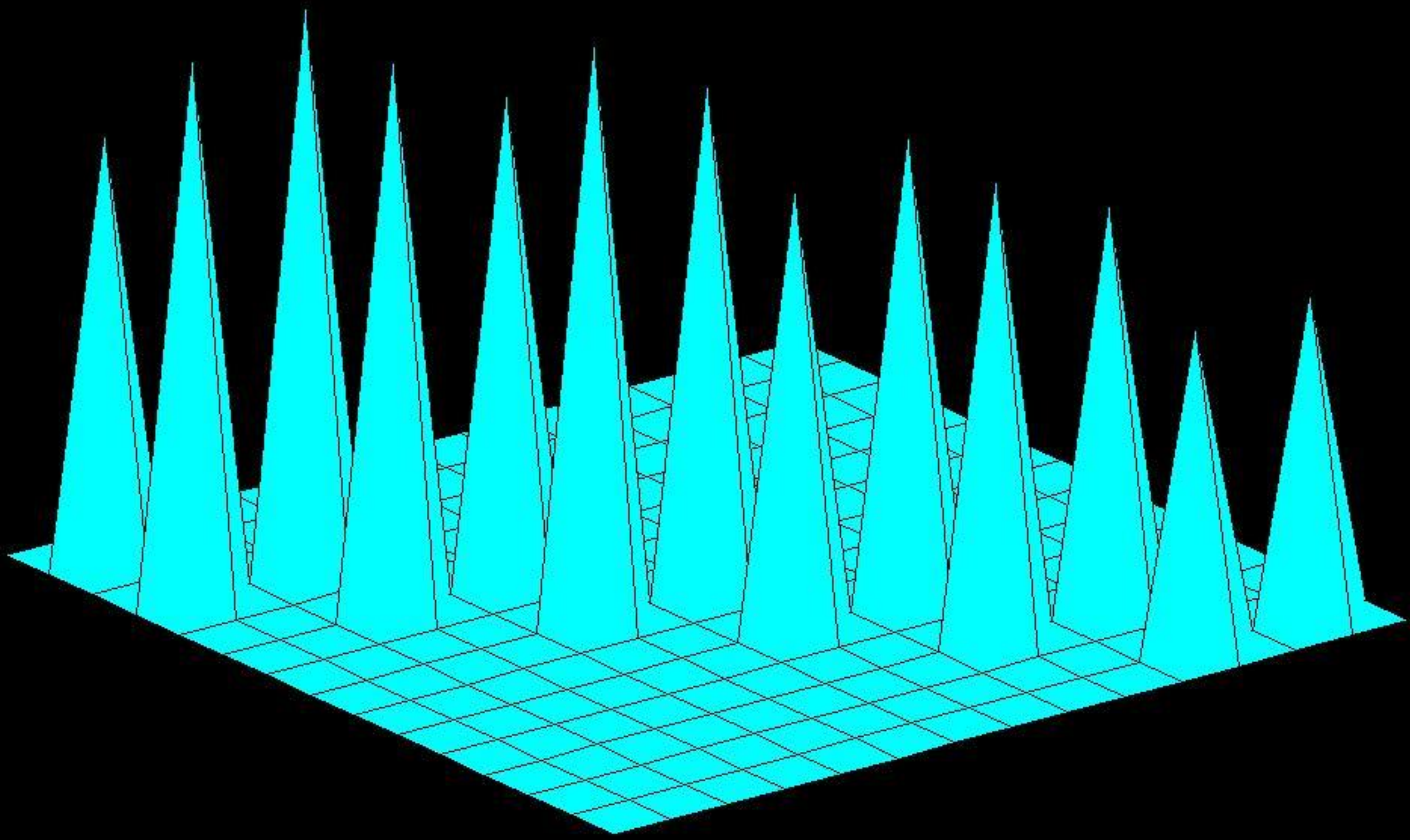
Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
4	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
5	DOE/SC/LBNL/NERSC United States	**Cori** - Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect Cray Inc.	622,336	14,014.7	27,880.7	3,939

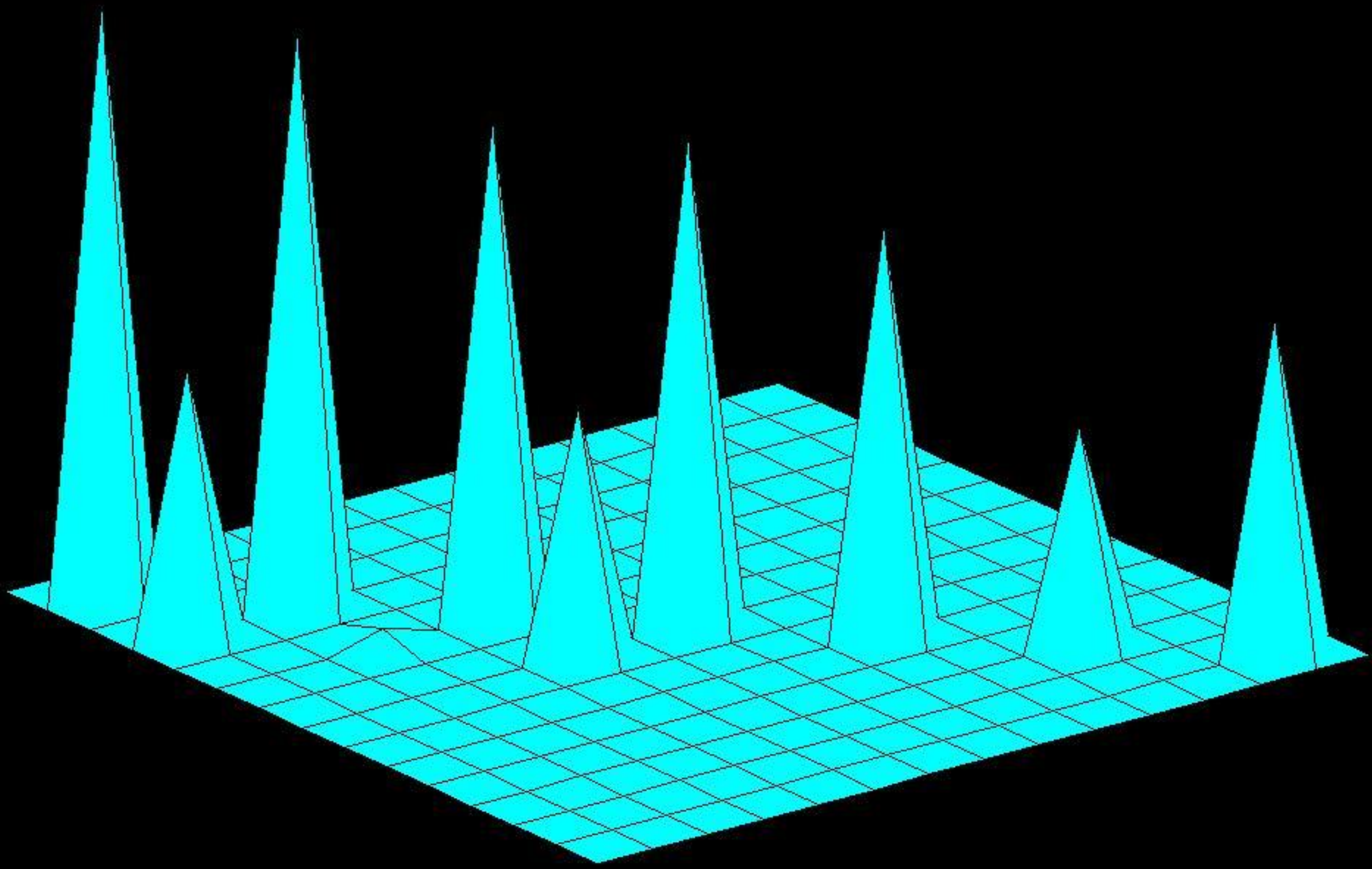
1979

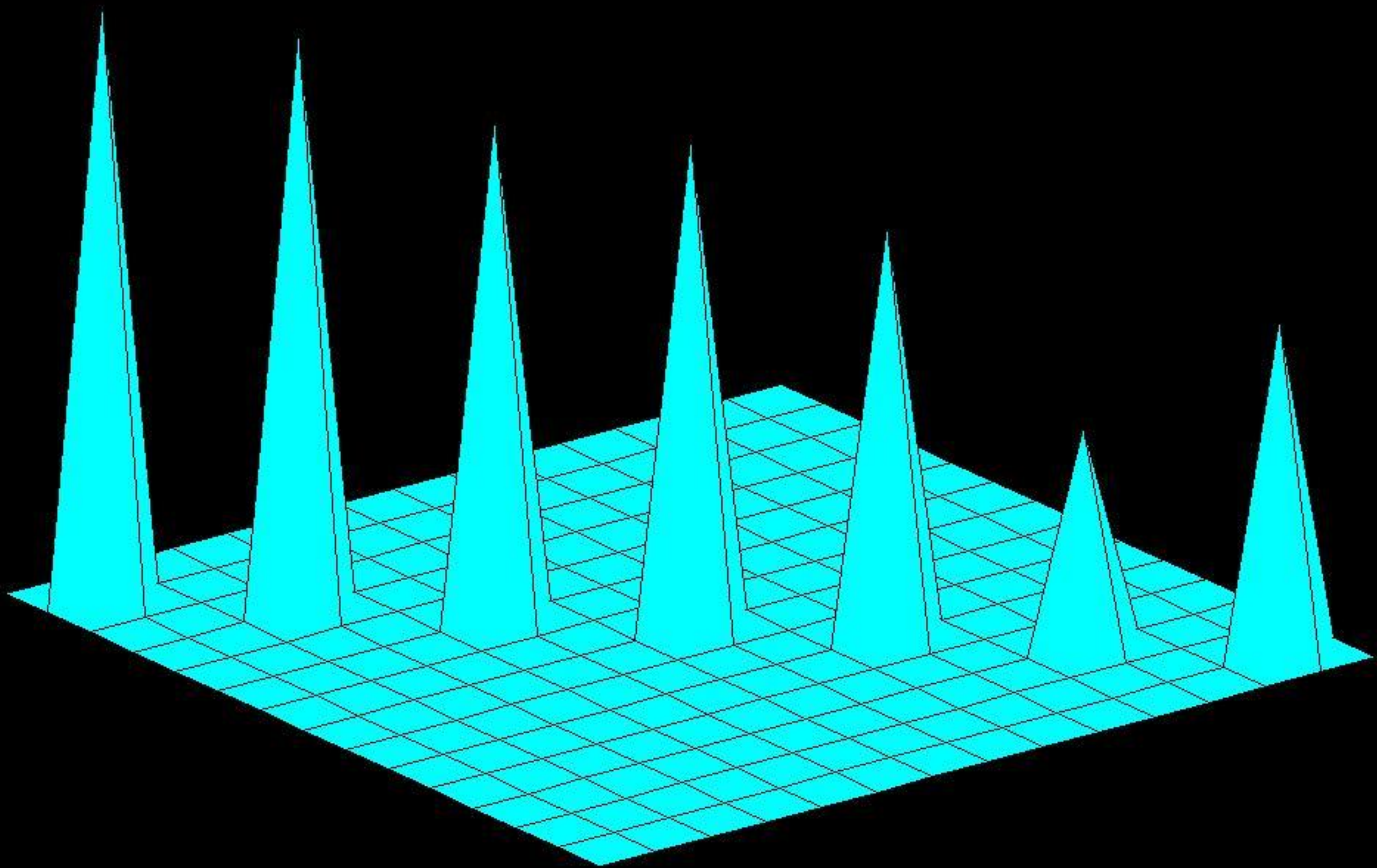

STARTREK



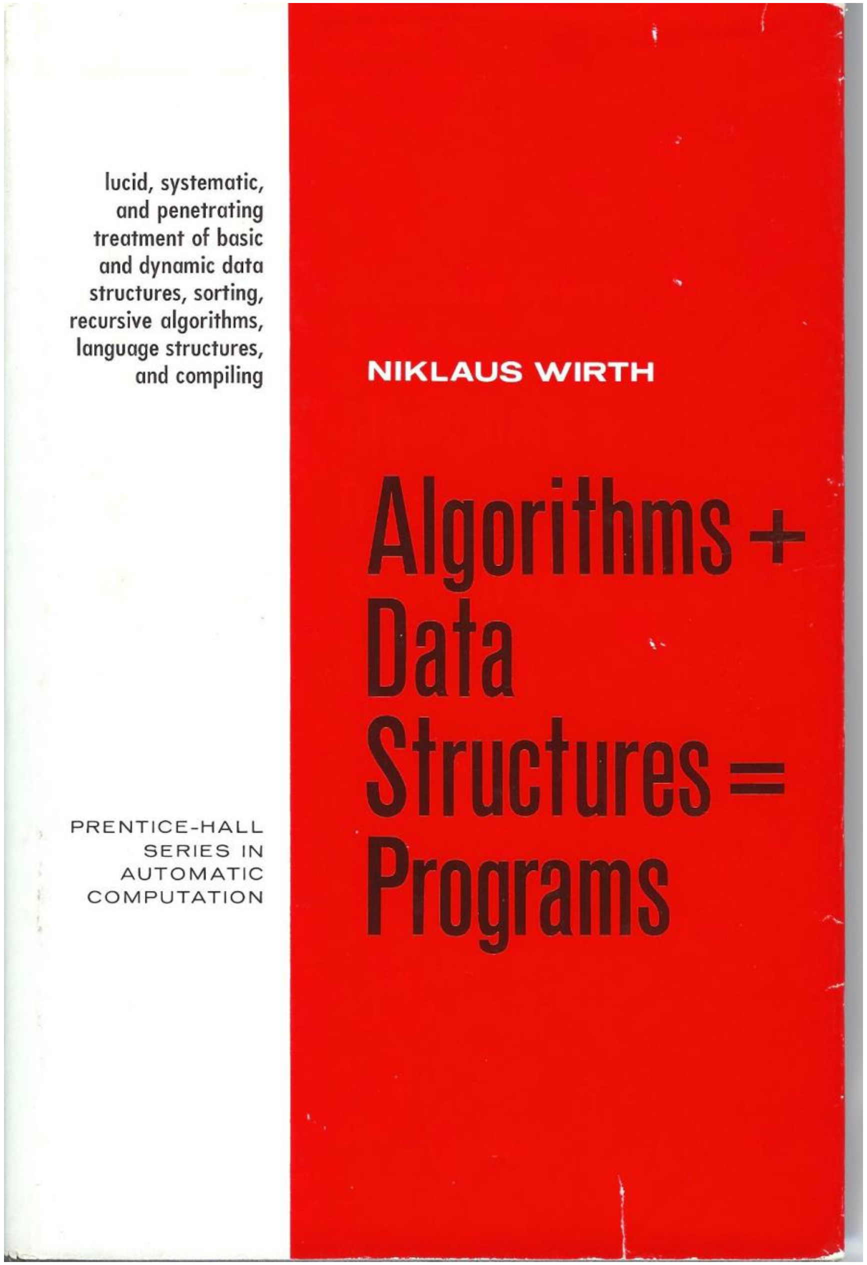








1977
MATLAB
Matrix Laboratory
PL/0
Fortran
Matrix data type



```
        < M A T L A B >
       Version of 01/10/81

HELP is available

<>help

Type HELP followed by
INTRO   (to get started)
NEWS    (recent revisions)
ABS   ANS   ATAN  BASE  CHAR  CHOL  CHOP  CLEA  COND  CONJ  COS
DET   DIAG  DIAR  DISP  EDIT  EIG   ELSE  END   EPS   EXEC  EXIT
EXP   EYE   FILE  FLOP  FLPS  FOR   FUN   HESS  HILB  IF    IMAG
INV   KRON  LINE  LOAD  LOG   LONG  LU    MACR  MAGI  NORM  ONES
ORTH  PINV  PLOT  POLY  PRIN  PROD  QR    RAND  RANK  RCON  RAT
REAL  RETU  RREF  ROOT  ROUN  SAVE  SCHU  SHOR  SEMI  SIN   SIZE
SQRT  STOP  SUM   SVD   TRIL  TRIU  USER  WHAT  WHIL  WHO   WHY
< > ( ) = . , ; \ / ' + - * :

<>
```

```
<> n = 10;
<> J = 1:n;
<> J = J(ones(n,1),:);
<> I = J';
<> E = ones(n,n);
<> A = E./(I-J+E/2);
<> long
<> S = svd(A)

S     =

  3.141592653589681
  3.141592653566663
  3.141592651393167
  3.141592527498736
  3.141587781570560
  3.141459305862260
  3.138952480600910
  3.104107683136387
  2.786915482404130
  1.300969070029703

<>
```

```
<>plot(x,x.*sin(3*x))

                                                         ****
                                                        *    *
                                                       *
                                                              *
                               ****                   *
                              *    **                          *
                             *                       *
                            *        *
     ******                *          *             *           *
*****      **             *             *
             **          *                         *             *
               *        *              *
                ***   **                          *                          *
                   ***                  *                         *
                                         *       *
                                          *     *                  *        *
                                           *   *
                                            ***
                                                                    *      *
                                                                     *
                                                                          *
                                                                      ****
```

Early Days at Argonne, Speakeasy

```
:_a=matrix(2,2:1,2,3,4) ; a
  A (A 2 by 2 Matrix)
  1 2
  3 4
:_a*a
  A*A (A 2 by 2 Matrix)
  7   10
  15  22
:_a/a
  A/A (A 2 by 2 Matrix)
  1  0
  0  1
:_aa=array(2,2:1,2,3,4)
:_aa*aa
  AA*AA (A 2 by 2 Array)
  1  4
  9  16
:_aa/aa
  AA/AA (A 2 by 2 Array)
  1  1
  1  1
```

Early Days at Argonne, Tek 4081

Cricket Lecture at Argonne Picnic

1979
Stanford

1979-80 Winter Quarter

CS237

Numerical Analysis

Math & CS Students

    Not impressed

Engineering Students

    Love MATLAB

1981

Commercial software for CDA

Integrated Systems Inc. (ISI)

Matrix-X

Systems Control Technology (SCT)

Ctrl-C

Jack Little

1984

MATLAB reimplemented in C
Jack Little and Steve Bangert
The MathWorks founded
Commercial MATLAB debut

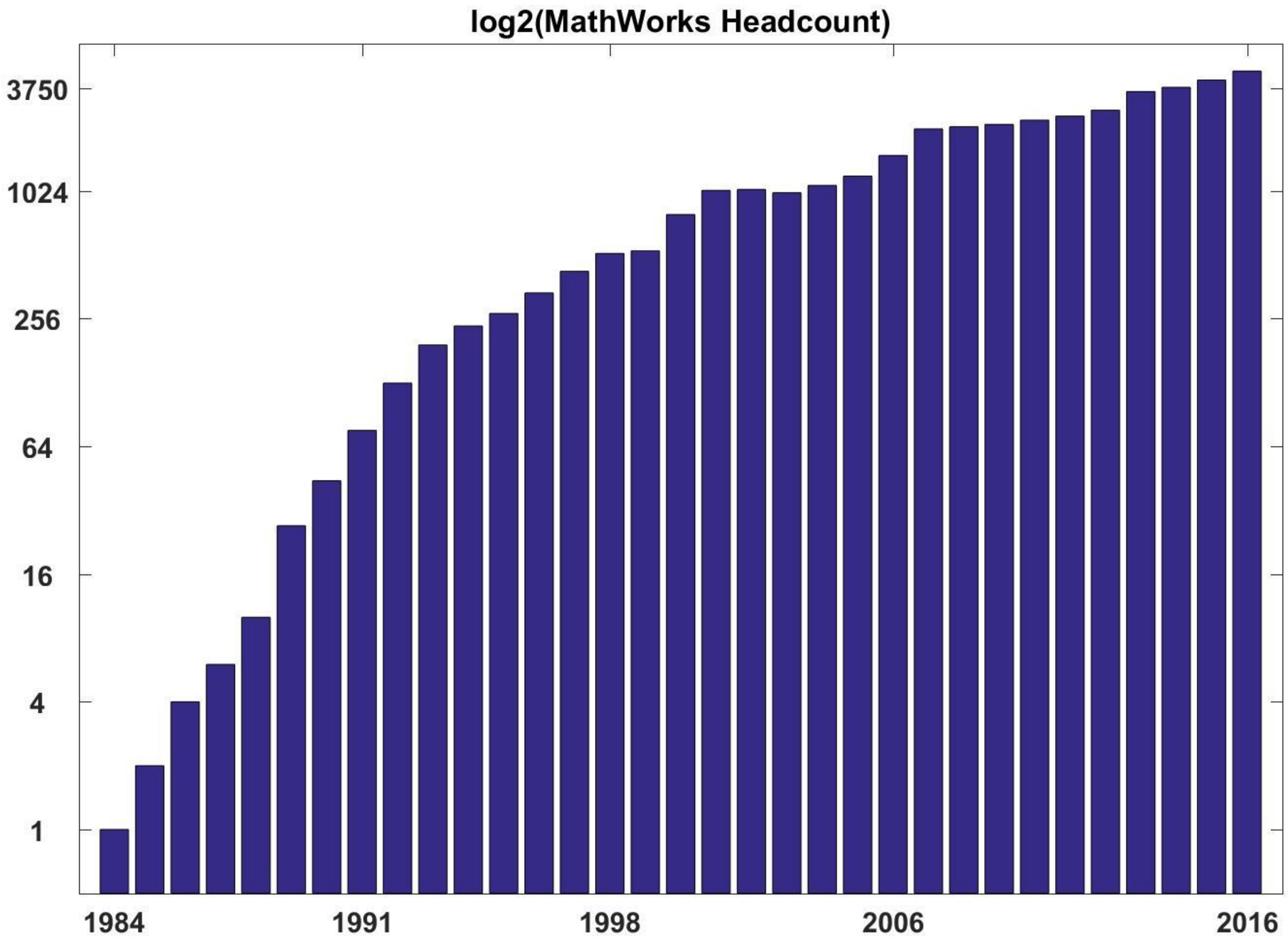
Jack's Compaq

MathWorks Headcount

1984-91

Doubles every year

1992-2014 Not so fast


log2(MathWorks Headcount)
3750
1024
256
64
16
4
1
1984
1991
1998
2006
2016

1987

1989

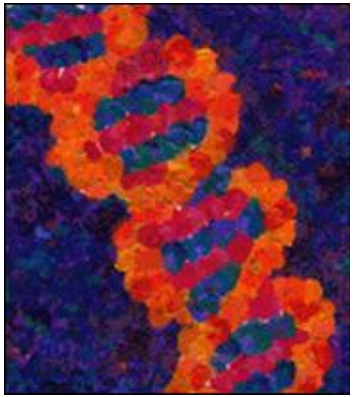
2014

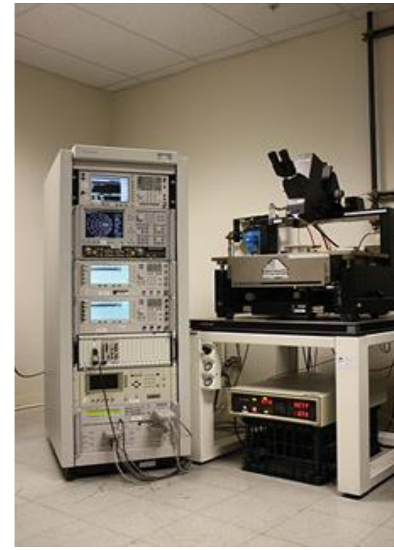

© Peter Sandberg, 2014

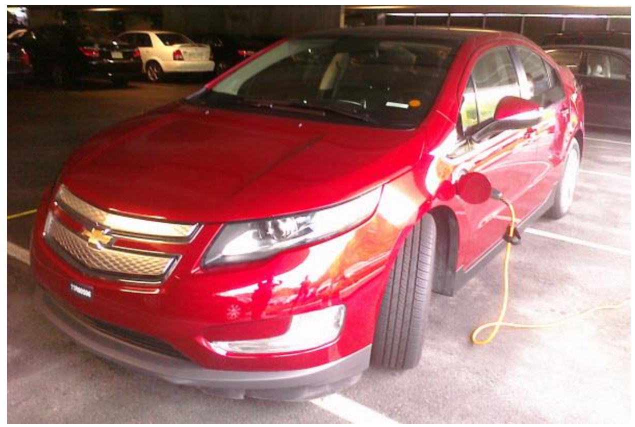
MathWorks at a Glance

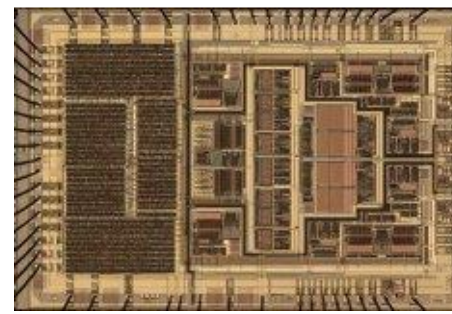
Earth's topography on a Miller cylindrical projection, created with MATLAB and Mapping Toolbox

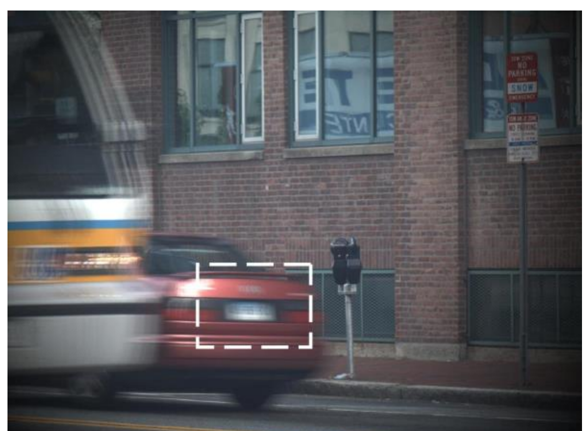
- **Headquarters:**
  Natick, Massachusetts US
- **Other U.S. Locations:**
  California, Michigan, Texas, Washington, DC
- **Europe:**
  France, Germany, Italy, Netherlands, Spain, Sweden, Switzerland, United Kingdom
- **Asia-Pacific:**
  Australia, China, India, Japan, Korea
- Worldwide training and consulting
- Distributors serving more than 20 countries







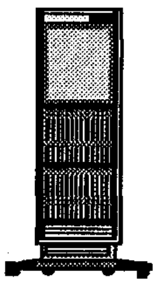


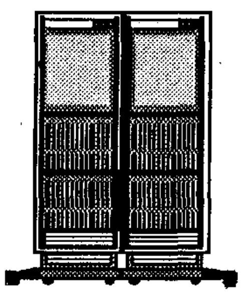


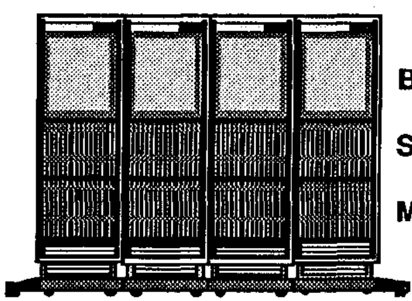
MATLAB and HPC

1985
Intel iPSC

The iPSC System Family





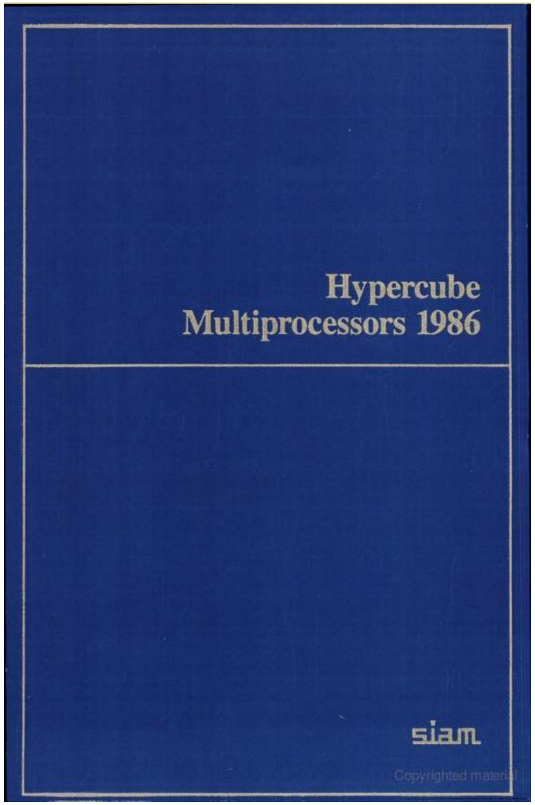


	Name	Nodes	Memory	MFLOPS	Price
Base System	iPSC/d5	32	16 MBytes	2	$171.5K
Memory System	iPSC/d4-MX	16	72 MBytes	1	$184.4K
Numeric System	iPSC/d4-VX	16	24 MBytes	106	$296.1K
Base System	iPSC/d6	64	32 MBytes	4	$293.5K
Memory System	iPSC/d5-MX	32	144 MBytes	2	$311.3K
Numeric System	iPSC/d5-VX	32	48 MBytes	212	$516.7K
Base System	iPSC/d7	128	64 MBytes	8	$524.6K
Symbolic System	iPSC/d6-MX	64	288 MBytes	4	$558.2K
Memory System	iPSC/d6-VX	64	96 MBytes	424	$947.5K

MATLAB on the iPSC

MATLAB runs only on PC front end.
Little support on iPSC itself.

1986



Mike Heath, editor,
“Proceedings of the First Conference on Hypercube Multiprocessors Knoxville, Tennessee, 1986."

“Embarrassingly Parallel”

One important way in which LINPACK and EISPACK will be used on such machines is in applications with many tasks involving matrices small enough to be stored on a single processor. The conventional subroutines can be used on the individual processors with no modification. We call such applications “embarrassingly parallel” to emphasize the fact that, while there is a high degree of parallelism and it is possible to make efficient use of many processors, the granularity is large enough that no cooperation between the processors is required within the matrix computations.

"Megaflops per Gallon"

iPSC Failure

Hardly any operating system.
Not enough memory.
No apps.
Parallelization hard, especially in 1986.

MathWorks

1990 - 2004

For 15 years, I am barely involved in HPC, except for …

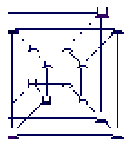
Why there isn't a parallel MATLAB

Our experience has made us skeptical

by Cleve Moler

There actually have been a few experimental versions of MATLAB for parallel computers. None of them has been effective enough to justify development beyond the experimental prototype. But we have learned enough from these experiences to make us skeptical about the viability of a fully functional MATLAB running on today's parallel machines. There are three basic difficulties:

- Memory model
- Granularity
- Business situation



A 16-node hypercube parallel computer. Each node can send messages directly to its nearest neighbors and indirectly to all other nodes.

Memory model

The most important attribute of a parallel computer is its memory model. Large-scale, massively parallel computers have potentially thousands of processors and *distributed memory*, that is, each processor has its own memory. Smaller scale machines, including some high-end workstations, have only a few processors and *shared memory*.

A good example of a distributed memory parallel computer is one of the first commercially available parallel computers, the *Intel iPSC*, where we tried to make our first parallel MATLAB almost ten years ago. It had up to 128 nodes—each a separate single board computer with an Intel microprocessor and maybe half a megabyte of memory. In principle, each node could execute a different program, but we usually ran the same program on all of them. Each node could send messages directly to its nearest neighbors and indirectly to all the other nodes. The whole machine was controlled by a front-end host, which initiated tasks, collected results, and handled all I/O.

We ran MATLAB on the host and gave names with capital letters to the functions in the parallel math library. So INV(A) or FFT(X) would start with a matrix in the host memory, split it into equally sized submatrices, send each of the submatrices to a node, invoke the parallel routine, and then collect the results back on the host. It took far longer to distribute the data than it did to do the computation. Any matrix that would fit into memory on the host was too small to make effective use of the parallel computer itself.

The situation hasn't changed very much in ten years. MATLAB is a lot bigger, and parallel computers are a lot faster. But distributed memory is still a fundamental difficulty. One of MATLAB's most attractive features is its memory model. There are no declarations or allocations—it is all handled automatically. The key question is: *Where are the matrices stored?* It is still true today that any matrix that fits into the host memory should probably stay there.

Granularity

A little over five years ago, we had a parallel MATLAB on a shared memory multiprocessor, the Ardent Titan, but we didn't tell the world about it. The most effective use of this machine, as well as today's multiprocessor workstations, is already done automatically by the operating system. MATLAB should run on only one processor, while other tasks, like the X-Windows server, use the other processors. In typical use, MATLAB spends only a small portion of its time in routines that can be parallelized, like the ones in the math library. It spends much more time in places like the parser, the interpreter, and the graphics routines, where any parallelism is difficult to find.

There are some special situations where parallel computation within MATLAB would be effective. For example, suppose I want to find what fraction of a large number of matrices have eigenvalues in the left half plane. The obvious place to parallelize this is on the outer loop. It's not necessary to use more than one processor to generate a single matrix or to compute its eigenvalues. The only place the processors would need to cooperate is in merging their final counts. However, to get MATLAB to handle this kind of parallelism would require fundamental changes to its architecture.

Business situation

It doesn't make good business sense for us to undertake fundamental changes in MATLAB's architecture. There are not enough potential customers with parallel machines. Most of the MATLAB community would rather see us devote our efforts to improving our conventional, uniprocessor software. So, we will continue to track developments in parallel computing, but we don't expect to get seriously involved again in the near future. ■

Cleve Moler is chairman and co-founder of The MathWorks. His e-mail address is moler@mathworks.com

Cleve's Corner, 1995
Why there isn't a parallel MATLAB

- Memory model
- Granularity
- Business situation

2004 + 2005
MATLAB returns to HPC

SC04, Pittsburgh
  Parallel MATLAB debut

SC05, Seattle
  Bill Gates keynote
  MATLAB demo

2015

MATLAB and Simulink
Many functions "parallel enabled".

Parallel Computing Toolbox (PCT)
Explicit parallelism at several levels.

Distributed Computing Server (MDCS)
Clusters and job managers.

Parallel Enabled

Simulink
Code Generation
Computational Biology
Control System Design and Analysis
Image Processing and Computer Vision
Math, Statistics, and Optimization
Signal Processing and Communications
Verification, Validation, and Test

Parallel Computing Toolbox

```
parpool
parfor
parfeval
spmd
batch
distributed
gpuArray
```

`gpuArray`

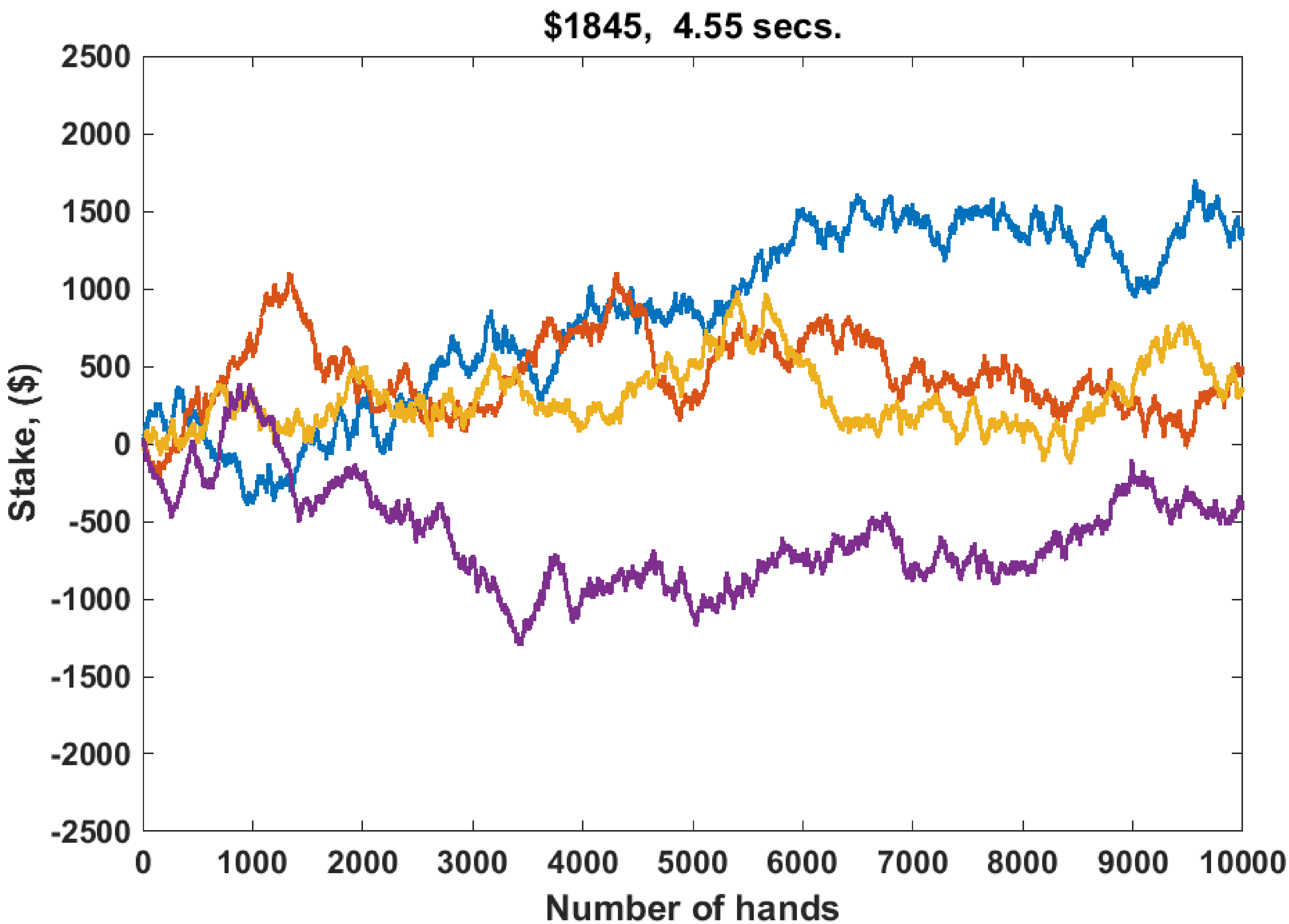
More popular than **`distributed`**.

Close cooperation with NVIDIA and
Univ. Tennessee ICL.

```
>> length(methods('gpuArray'))
   ans =
   399
```

```
% BLACKJACKDEMO  Parallel blackjack demo.

p = 4;       % Number of players.
n = 10000;   % Number of hands per player.
B = zeros(n,p);
tic
parfor k = 1:p
   B(:,k) = blackjack(n);
end
t = toc;
s = sum(B(n,:));
plot(B)
```


$1845, 4.55 secs.
2500
2000
1500
1000
500
0
-500
-1000
-1500
-2000
-2500
Stake, ($)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Number of hands

```
>> blackjackdemo
Elapsed time is 8.136478 seconds.

>> parpool
Starting parallel pool (parpool) using the
  'local' profile ... connected to 2 workers.

>> blackjackdemo
Elapsed time is 4.389613 seconds.
```

```
% BLACKJACKDEMO  Parallel blackjack demo.

p = 4;         % Number of players.
n = 10000;   % Number of hands per player.
B = zeros(n,p);
tic
parfor k = 1:p
   B(:,k) = blackjack(n);
end
toc
plot(B)
```

MATLAB HPC Today

Dominant use: embarrassingly parallel **`parfor`**.

Distributed arrays: useful data structure.

As far as we know: little distributed dense
linear algebra.

MATLAB HPC Today

Job managers rule:
    File systems, security, privacy

Shared facilities preclude interactivity:
    Argonne's Jazz

MATLAB Today

- Simulink
- Stateflow
- Dozens of toolboxes & blocksets
- Thousands of functions
- Thousands of pages of documentation
- Several million lines of code
  - Almost half M
  - Almost half C
  - Some Java
  - Some Fortran

MATLAB's historical and intellectual basis is numerical linear algebra.

MATLAB's commercial success derives from applications in technical computing.

"The reason MATLAB is so good at signal processing is that it was not designed for signal processing. It was designed to do mathematics."

-- Jim McClellan
Georgia Tech

MATLAB

Not just “Matrix Laboratory” anymore.

On the Web

//www.mathworks.com/products/
parallel-computing/
matlab-parallel-cloud

//www.mathworks.com/matlabcentral

//www.mathworks.com/moler

//blogs.mathworks.com/cleve