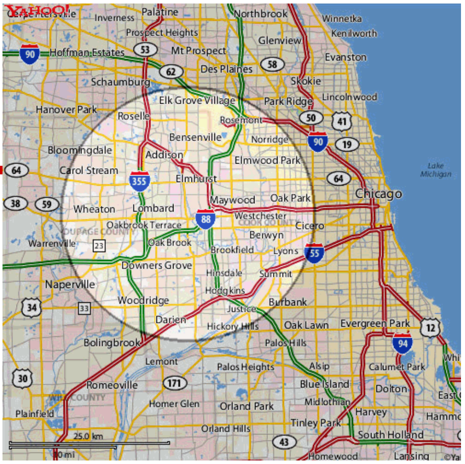
STILL HAVING FUN AFTER 50 YEARS
SOME OF JACK’S GREATEST HITS

Jack Dongarra
University of Tennessee
Oak Ridge National Laboratory
University of Manchester

How Did I Get Here?



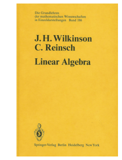
- Grew up on the southwest side of Chicago.
- Wanted to be a high school science teacher.
- Attending the “normal school” in Chicago that trained teachers, Chicago State University.
- Took many math and science courses.
- The last year in college everything changed.

Argonne's Undergraduate Research Program

- In the Fall of 1971 the Chair of the Department of Physics at Chicago State University, Harvey Left, suggested I apply to the Argonne's Undergraduate Research Program.
    - About 50 undergraduates from around the country are selected to come to ANL for a semester.
    - What I didn't know was that Harvey Left was a college roommate with Jim Pool who was then the head of Applied Math Division at Argonne.
  - So, my last semester at Chicago State was spent as an intern at Argonne.
- When I started at Argonne I had no formal training in computer science with very little programming experience.
  - I was assigned to work with Brian Smith who was leading the EISPACK effort, and I quickly tried to learn all about linear algebra and eigenvalue problems.
  - I realized that computer science was fun and is what I wanted to pursue.
- Applied to Illinois Institute of Technology for a Master's degree in CS.
  - While working on Master's at IIT worked at Argonne one day a week on EISPACK.
  - Wasn't sure I wanted a PhD.
  - When I graduated from IIT with an MS Jim Pool offered me a job at Argonne.

< M A T L A B >
VERSION OF 07/29/80
HELP is available
<>
exec('x')
open file X , unit = 8
<>
a=<'LINPACK'
'INPACK'
'NPACK'
'PACK'
'ACK'
'CK'
'K'>
ANS =
169.8649+ 0.0000*i
rcond(a)
ANS =
0.0356
348 flops

Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented

NAG Was a Big Influence @ ANL and Me

- Argonne's MCS Division had a close relationship with NAG.
- My first overseas trip was to visit NAG in 1975 at the invitation of Brian Ford.
- Met up with Wilkinson at National Physical Laboratory and first met Sven.

- Stopped in at Harwell and visited with Iain and John.

My First Paper

SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 9, 219–226 (1979)

Unrolling Loops in FORTRAN*

J. J. DONGARRA AND A. R. HINDS
Argonne National Laboratory, Argonne, Illinois 60439, U.S.A.

SUMMARY

The technique of 'unrolling' to improve the performance of short program loops resorting to assembly language coding is discussed. A comparison of the benefit 'unrolling' on a variety of computers using an assortment of FORTRAN com presented.

- Reduces loop overhead
  - Level of unrolling dedicated by the instruction stack size
- Help the compiler to:
  - Facilitates pipelining
  - Increases the concurrence between independent functional units
- Provided ~15% performance improvement

TECHNIQUE

When a loop is unrolled, its contents are replicated one or more times, with appropriate adjustments to array indices and the loop increment. For instance, the DAXPY[9] sequence, which adds a multiple of one vector to a second vector:

```
   DO 10 I = 1,N
      Y(I) = Y(I)+A*X(I)
10 CONTINUE
```

would, unrolled to a depth of four, assume the form

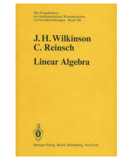
```
   M = N-MOD(N,4)
   DO 10 I = 1,M,4
      Y(I)   = Y(I)   +A*X(I)
      Y(I+1) = Y(I+1)+A*X(I+1)
      Y(I+2) = Y(I+2)+A*X(I+2)
      Y(I+3) = Y(I+3)+A*X(I+3)
10 CONTINUE
```

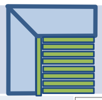
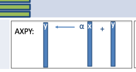
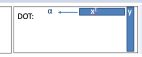
Basic Linear Algebra Subprograms for Fortran Usage

C. L. LAWSON
Jet Propulsion Laboratory
R. J. HANSON
Sandia Laboratories
D. R. KINCAID
The University of Texas at Austin
and
F. T. KROGH
Jet Propulsion Laboratory

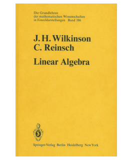
A package of 38 low level subprograms for many of the basic operations of numerical linear algebra is presented. The package is intended to be used with Fortran. The operations in the package include dot product, elementary vector operation, Givens transformation, vector copy and swap, vector norm,

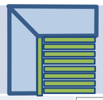
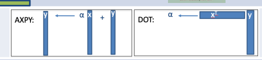
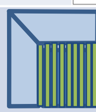
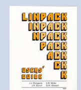
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)	 	Standards for: Vector-Vector operations

Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)		Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)	 	Rely on - Level-1 BLAS operations - Column oriented

An Accidental Benchmarker

Top 23 List from 1977

- Appendix B of the Linpack Users' Guide
  - Designed to help users extrapolate execution time for Linpack software package
- First benchmark report from 1977;

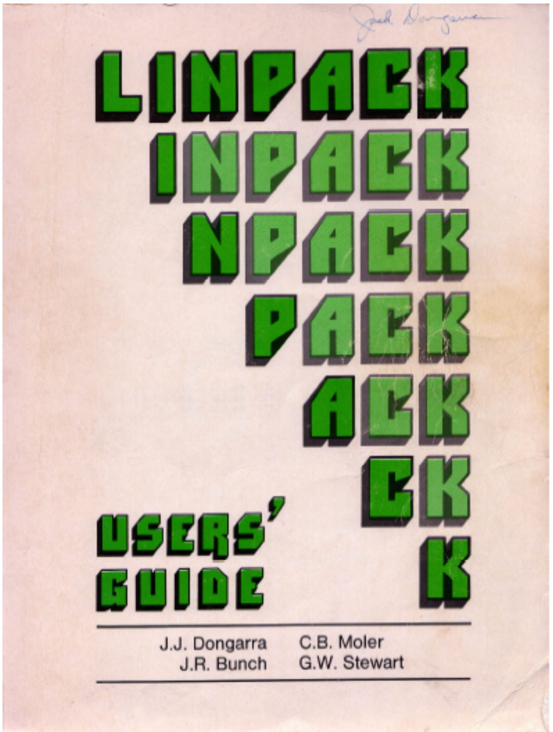
LINPACK was an NSF Project w/ ANL, UNM, UM, & UCSD

Worked independently and came to Argonne in the summers

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

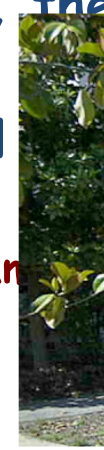
UNIT = 10**6 TIME/(1/3 100**3 + 100**2)

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



Late 70's - New Mexico Days

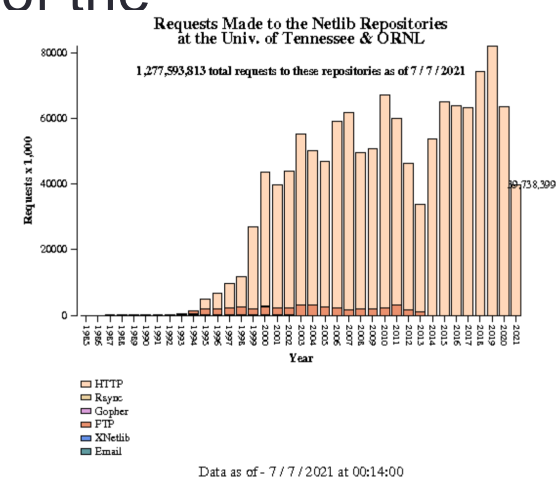
- Encouraged to pursue PhD by many visitors.
- Cleve said he would customize a degree program at the U of New Mexico in the Math Department.
- I was detailed from Argonne to work at Los Alamos (LANL).
  - Programmed the first Cray-1's.
    - Found a bug in the Cray-1 HW with the Linpack Ben
- Spent one semester at UNM@LANL, the
  2 semesters on UNM campus.
- Cleve was at Stanford on Sabbatical
  my last year at UNM.
  - The plan was to finish my courses & exan
    and then join Cleve at Stanford.
- On to Stanford and Serra House.



Netlib Mathematical Software & More

Over 1.2 B Served

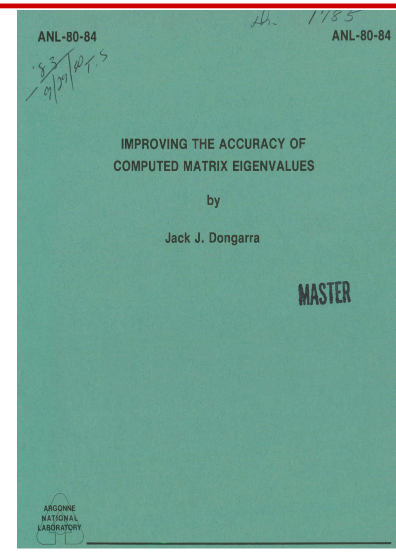
- Began in 1984
  - Eric Grosse @ Bell Labs and JD @ ANL
  - Motivated by need for cost-effective, timely distribution of high-quality mathematical software to the community.
- One of the first open source software collections.
- Designed to send, by return electronic mail, requested items. Email to netlib@mcs.anl.gov an in the body of the email: `Send dgefa from linpack`



- Moderated collection /Distributed maintenance
- Also mail forwarding in 1983 & na-digest started in 1987.

Exploiting Mixed Precision

- Wilkinson gave a lecture at ANL on the topic of improving λ's.
- My interest in mixed precision began with my dissertation …
  - *Improving the Accuracy of Computed Matrix Eigenvalues*
    - Compute the eigenvalues and eigenvector in low precision then improve selected values/vectors to higher precision for $O(n^2)$ ops using the the matrix decomposition
  - Extended to singular values, 1983
  - Algorithm in TOMS 710, 1992



SIAM J. NUMER. ANAL.
Vol. 20, No. 1, February 1983

IMPROVING THE ACCURACY OF COMPUTED EIGENVALUES AND EIGENVECTORS*

J. J. DONGARRA,† C. B. MOLER‡ AND J. H. WILKINSON§

Abstract. This paper describes and analyzes several variants of a computational method for improving the numerical accuracy of, and for obtaining numerical bounds on, matrix eigenvalues and eigenvectors. The method, which is essentially a numerically stable implementation of Newton's method, may be used to "fine tune" the results obtained from standard subroutines such as those in EISPACK [Lecture Notes in Computer Science 6, 51, Springer-Verlag, Berlin, 1976, 1977]. Extended precision arithmetic is required in the computation of certain residuals.

Introduction. The calculation of an eigenvalue λ and the corresponding eigenvector x (here after referred to as an eigenpair) of a matrix A involves the solution of the nonlinear system of equations

$$(A-\lambda I)x=0.$$

WHY MIXED PRECISION?

- There are many reasons to consider mixed precision in our algorithms…
  - Less Communication
    - Reduce memory traffic
    - Reduce network traffic
  - Reduce memory footprint
  - More Flop per second
    - Reduced energy consumption
    - Reduced time to compute
  - Accelerated hardware in current architecture.
  - Suitable numerical properties for some algorithms & problems.

Did A Brief Postdoc at IBM Watson – IJK Paper

SIAM REVIEW
Vol. 26, No. 1, January, 1984

0036–1445/84/2601–0003 $01.25/0

IMPLEMENTING LINEAR ALGEBRA ALGORITHMS FOR DENSE MATRICES ON A VECTOR PIPELINE MACHINE*

J. J. DONGARRA,† F. G. GUSTAVSON‡ AND A. KARP§

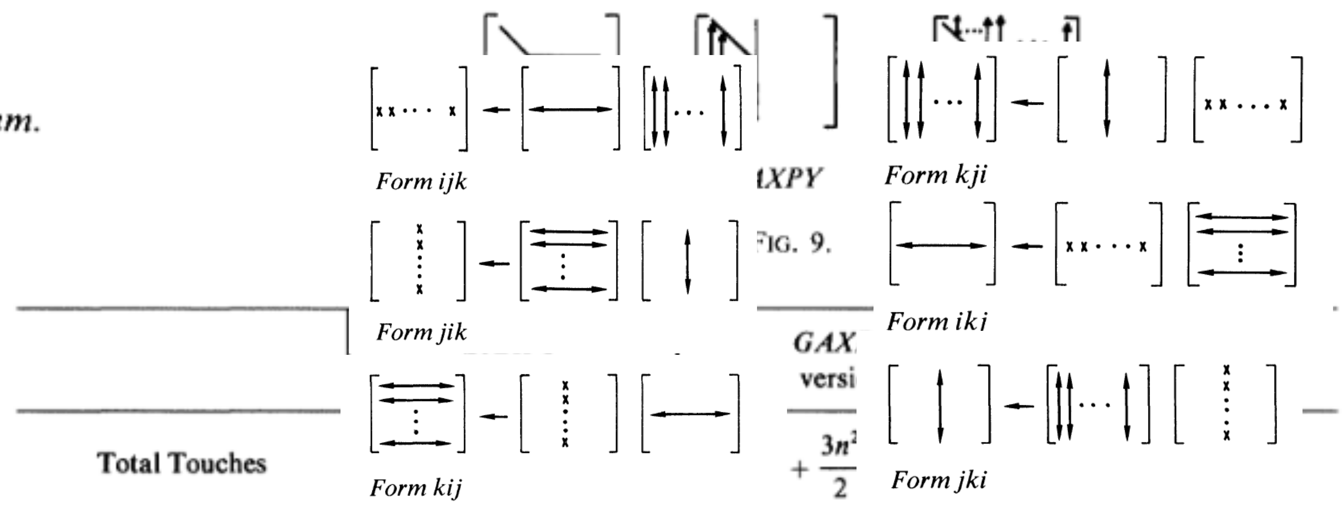
Abstract. This paper examines common implementations of linear algebra algorithms, such as matrix-vector multiplication, matrix-matrix multiplication and the solution of linear equations. The different versions are examined for efficiency on a computer architecture which uses vector processing and has pipelined instruction execution. By using the advanced architectural features of such machines, one can usually achieve maximum performance, and tremendous improvements in terms of execution speed can be seen over conventional computers.

Generic matrix multiplication algorithm.

for _____ = 1 to _____
  for _____ = 1 to _____
    for _____ = 1 to _____
      $c_{ij} = c_{ij} + a_{ik} * b_{kj}$
    end
  end
end

Generic Gaussian elimination algorithm.

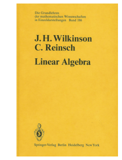
for ______________
  for ______________
    for ______________
      $a_{ij} = a_{ij} - (a_{ik} * a_{kj})/a_{kk}$.
    end
  end
end

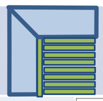
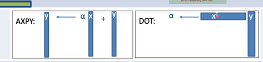
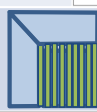
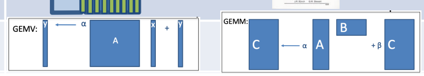


Argonne Math & CS Division 1986

Back row: Jim Boyle (w/picture of Larry Wos), John Gabriel, Ken Dritz, Joe Cook, Bob Veroff, Hans Kaper, Paul Messina, Bernie Matkowsky, Jim Cody, James Lyness, Wayne Cowell, Bert Garbow, Ken Hillstrom, Brian Smith, ? Seated: JD, Rusty Lusk, Mike Minkoff, Larry Leaf, Jorge More', Danny Sorensen, Bruce Char, Doris Pool, Judy Beumer

Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)		Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)	 	Rely on - Level-1 BLAS operations - Column oriented
Level 2 & 3 BLAS - ATLAS		Standards for: Matrix-Vector & Matrix-Matrix operations

An Extended Set of FORTRAN Basic Linear Algebra Subprograms

JACK J. DONGARRA
Argonne National Laboratory
JEREMY DU CROZ and SVEN HAMMARLING
Numerical Algorithms Group, Ltd.
and
RICHARD J. HANSON
Sandia National Laboratory

ACM Transactions on Mathematical Software, Vol. 14, No. 1, March 1988, Pages 1–17.

A Set of Level 3 Basic Linear Algebra Subprograms

JACK J. DONGARRA
University of Tennessee and Oak Ridge National Laboratory
JEREMY DU CROZ and SVEN HAMMARLING
Numerical Algorithms Group, Ltd.
and
IAIN DUFF
Harwell Laboratory

ACM Transactions on Mathematical Software, Vol. 16, No. 1, March 1990, Pages 1–17.

Summer's in East Hagbourne & Oxford









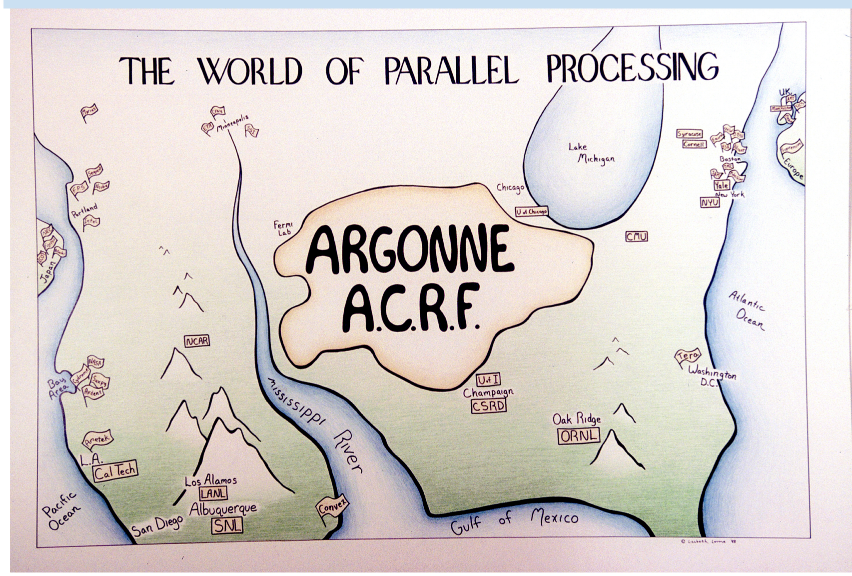
Argonne's Parallel Menagerie

ACTIVITIES AND OPERATIONS OF THE

ADVANCED COMPUTING RESEARCH FACILITY

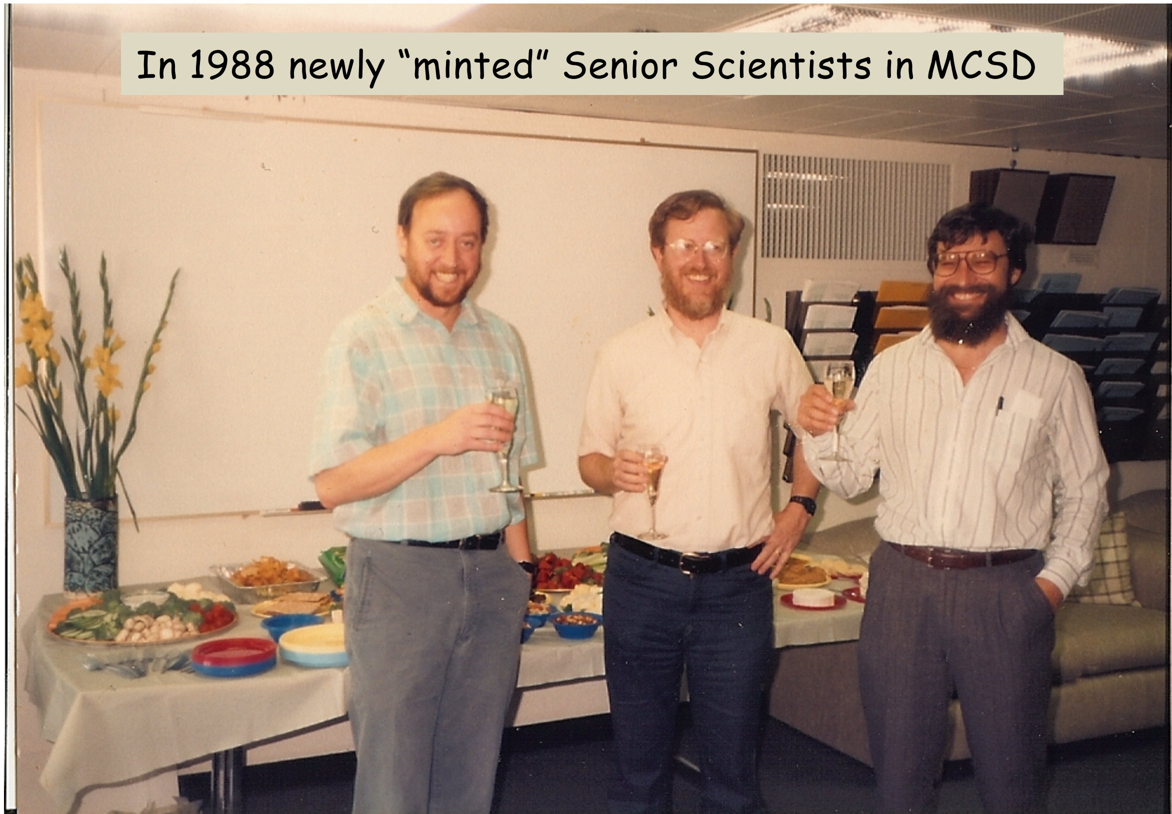
Several radically different parallel architectures, from shared to distributed memory; from vector to dataflow to bit parallel processors

- Thinking Machines CM-2, w/16,384 procs.
- Active Memory Technology DAP-510, w/1024 procs.
- BBN TC 2000 (Butterfly II), w/32 procs.
- Cydrom Cydra 5, VLIW architecture
- Denelcor HEP, w/4 PEMs
- Intel iPSC/d5 hypercube w/32 procs.
- Sequent Balance 21000, w/24 procs.
- Encore Multimax, w/20 procs.
- Intel iPSC/d4 hypercube, w/16 vector procs.
- Alliant FX/8, w/8 vector procs.
- Ardent Titan graphics supercomputer, w/4 vector procs.



SIAM J. SCI. STAT. COMPUT.
Vol. 8, No. 2, March 1987

In 1988 newly “minted” Senior Scientists in MCSD



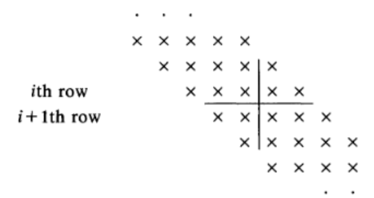
problem.
uting the
hat retains
ghout the
ot finding
hogonality
presented
all levels

ging with
hen run in
lems, and

divide a given
he symmetric

$$T=\begin{pmatrix} T_1 & \beta e_k e_1^T \\ \beta e_1 e_k^T & T_2 \end{pmatrix}=\begin{pmatrix} \hat{T}_1 & 0 \\ 0 & \hat{T}_2 \end{pmatrix}+\theta\beta\begin{pmatrix} e_k \\ \theta^{-1} e_1 \end{pmatrix}(e_k^T,\ \theta^{-1}e_1^T) \tag{2.1}$$

This algorithm can be generalized to handle band matrices as well. Instead of performing a rank-one tearing to split the matrix into two independent subproblems, we proceed by making $(m+1)m/2$ rank-one changes designed to split the matrix, (m is the half bandwidth, $m=1$ for tridiagonal matrices).



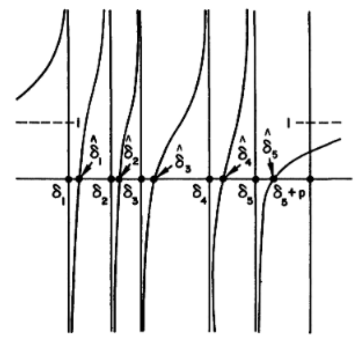


FIG. 1. *The secular equation.*

In 1989 Decided to Move to Tennessee

Bob Ward and Mike Heath made me an offer I couldn't refuse

Moved and setup shop: "Innovative Computing Laboratory"

Many Visitors in Tennessee as Well

Journal of Computational and Applied Mathematics 27 (1989) 215–227
North-Holland

Based on the block Householder work of Charlie VL and Chris B…

Block reduction of matrices to condensed forms for eigenvalue computations

Jack J. DONGARRA* and Danny C. SORENSEN**
Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, U.S.A.

Sven J. HAMMARLING
Numerical Algorithms Group Ltd., Wilkinson House, Jordan Hill Road, Oxford, United Kingdom OX2 8DR

Received 11 February 1988
Revised 7 October 1988

Abstract: In this paper we describe block algorithms for the reduction of a real symmetric matrix to tridiagonal form and for the reduction of a general real matrix to either bidiagonal or Hessenberg form using Householder transformations. The approach is to aggregate the transformations and to apply them in a blocked fashion, thus achieving algorithms that are rich in matrix–matrix operations. These reductions to condensed form typically comprise a preliminary step in the computation of eigenvalues or singular values. With this in mind, we also demonstrate how the initial reduction to tridiagonal or bidiagonal form may be pipelined with the divide and conquer technique for computing the eigensystem of a symmetric matrix or the singular value decomposition of a general matrix to achieve algorithms which are load balanced and rich in matrix–matrix operations.

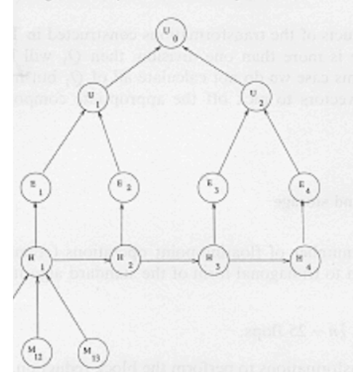
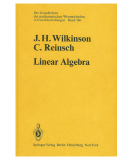
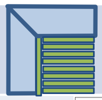
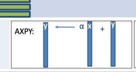
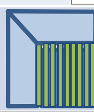
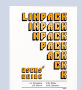
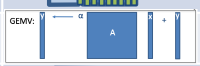
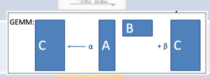
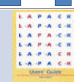


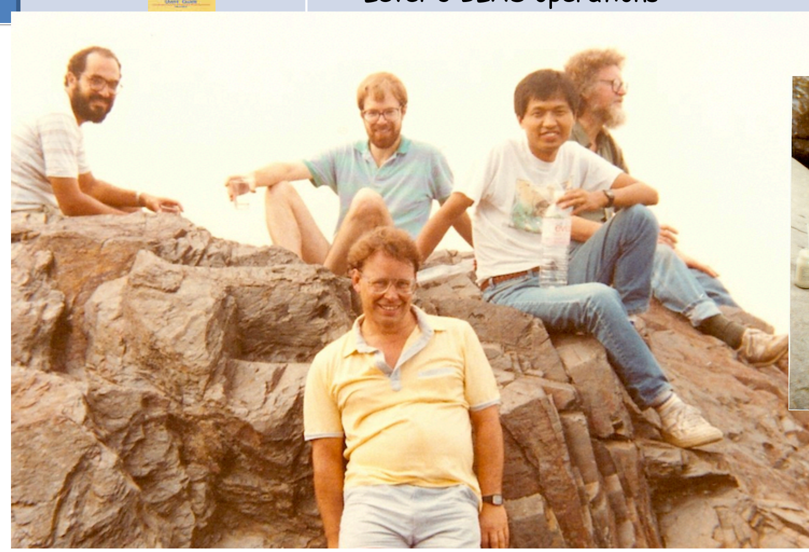
Fig. 2. Large-grain control flow graph.

Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)	 	Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)	 	Rely on - Level-1 BLAS operations - Column oriented
Level 2 & 3 BLAS - ATLAS	 	Standards for: Matrix-Vector & Matrix-Matrix operations
LAPACK (1990's) (Blocking, cache friendly)	 	Rely on - Level-3 BLAS operations

Missing C. Bischof, S. Blackford, A. McKinney, D. Sorensen



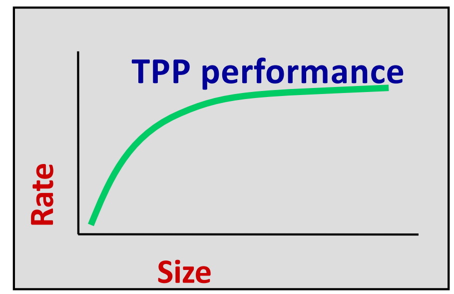
Top500 Since 1993

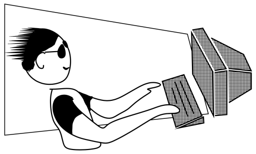
- Since 1978 I maintained a LINPACK Benchmark list.
- Hans Meuer and Erich S. had a list of fastest computers ranked by peak performance
- Listing of the 500 most powerful computers in the World
- Yardstick: Performance for

  Ax=b, dense problem

Maintained and updated twice a year

SC'xy in the States in November

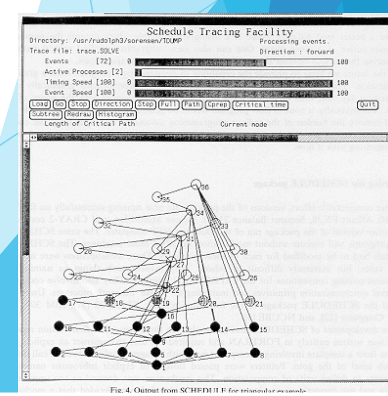




PVM

Parallel Computing was off and running, but no standards existed.

- Al Geist and Vaidy Sunderam started the project in the summer of 1989 at ORNL.
  - Version 1 was used internally at ORNL and not released.
- Liz Jessup came to ORNL as the Householder Fellow, July 1990 – July 1991
  - Bob Manchek
- Bob was hired by ICL to rewrite PVM and work on extending a DAG based programming system built on PVM called SCHEDULE.
- PVM was done by 6 people in the mountains of Tennessee.
- Many message passing systems in use.
- Clear a standard was needed.



ORNL/TM-12231

Engineering Physics and Mathematics Division

Mathematical Sciences Section

A PROPOSAL FOR A USER-LEVEL, MESSAGE-PASSING INTERFACE IN A DISTRIBUTED MEMORY ENVIRONMENT

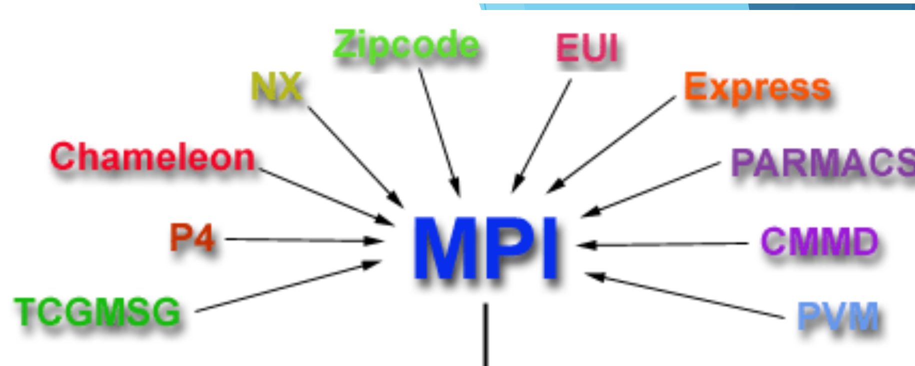
Jack J. Dongarra ‡§
Rolf Hempel ¶
Anthony J. G. Hey †
David W. Walker §

‡ Department of Computer Science
107 Ayres Hall
Knoxville, TN 37996-1301

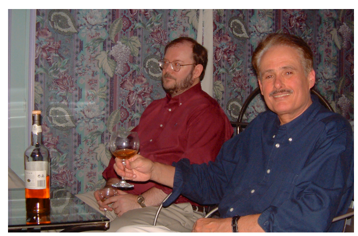
† Department of Electronics and Computer Sciences
University of Southampton
Southampton, SO9 5NH
United Kingdom

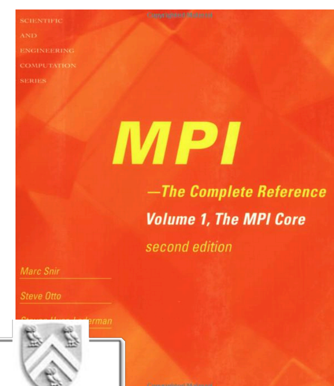
¶ Gesellschaft für Mathematik und Datenverarbeitung mbH
P. O. Box 1316
D-5205 Sankt Augustin 1
Germany

§ Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367



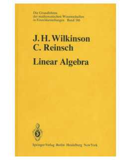
Draft report in November 1992
Many meeting involving the community
MPI standard introduced in 1994

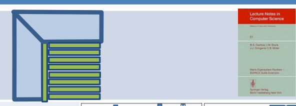
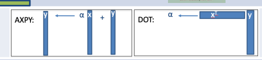
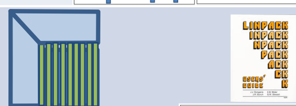
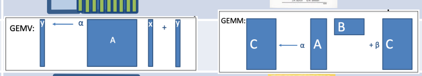
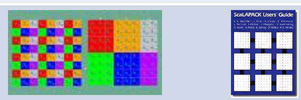




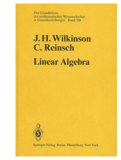
CRPC
The Center for Research on Parallel Computation at Rice University

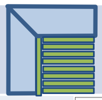
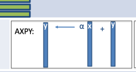
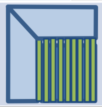
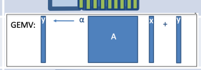
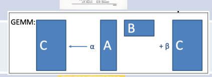
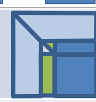
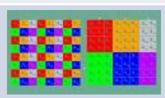
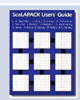
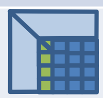
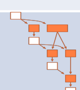
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



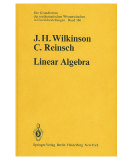
Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)		Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)		Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)		Rely on - Level-1 BLAS operations - Column oriented
Level 2 & 3 BLAS - ATLAS		Standards for: Matrix-Vector & Matrix-Matrix operations
LAPACK (1990's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
PVM and MPI		Standards for: Message passing
ScaLAPACK (2000's) (Distributed Memory)		Rely on - PBLAS Mess Passing

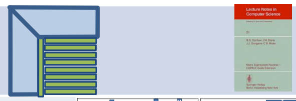
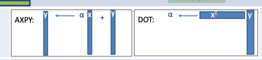
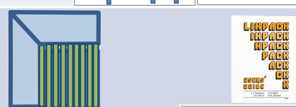
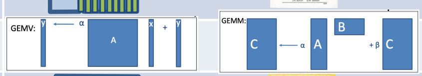
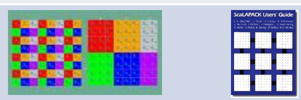
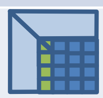
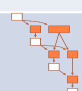
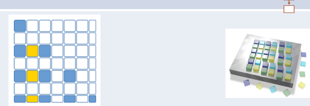
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments



Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)	 	Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)	 	Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)	 	Rely on - Level-1 BLAS operations - Column oriented
Level 2 & 3 BLAS - ATLAS	 	Standards for: Matrix-Vector & Matrix-Matrix operations
LAPACK (1990's) (Blocking, cache friendly)	 	Rely on - Level-3 BLAS operations
PVM and MPI		Standards for: Message passing
ScaLAPACK (2000's) (Distributed Memory)	 	Rely on - PBLAS Mess Passing
PLASMA / DPLASMA / MAGMA (2010's) (Many-core friendly & GPUs)		Rely on - DAG/scheduler - block data layout
PaRSEC		Standards for: Scheduling

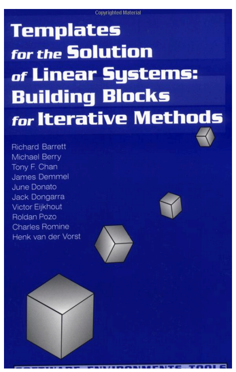
Over the Past 50 Years Evolving SW and Alg Tracking Hardware Developments

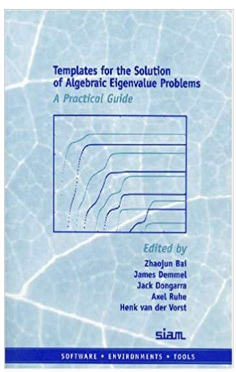


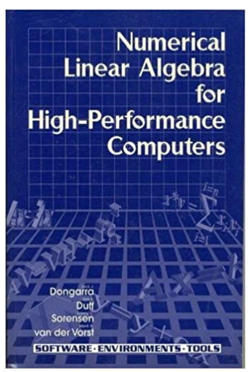
Features: Performance, Portability, and Accuracy		
EISPACK (1970's) (Translation of Algol to F66)		Rely on - Fortran, but row oriented
Level 1 Basic Linear Algebra Subprograms (BLAS)		Standards for: Vector-Vector operations
LINPACK (1980's) (Vector operations)		Rely on - Level-1 BLAS operations - Column oriented
Level 2 & 3 BLAS - ATLAS		Standards for: Matrix-Vector & Matrix-Matrix operations
LAPACK (1990's) (Blocking, cache friendly)		Rely on - Level-3 BLAS operations
PVM and MPI		Standards for: Message passing
ScaLAPACK (2000's) (Distributed Memory)		Rely on - PBLAS Mess Passing
PLASMA / DPLASMA / MAGMA (2010's) (Many-core friendly & GPUs)		Rely on - DAG/scheduler - block data layout
PaRSEC		Standards for: Scheduling
SLATE (2020's) (DM and Heterogeneous arch)		Rely on C++ - Tasking DAG scheduling - Tiling, but tiles can come from anywhere - Heterogeneous HW, Batched dispatch

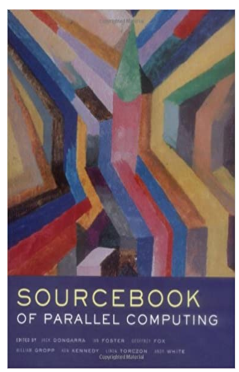
Community Efforts

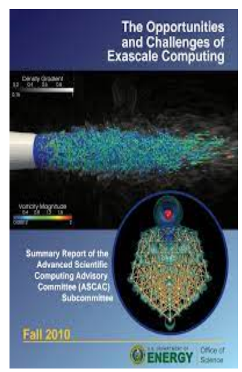
- BLAS
- MPI
- Top500
- International Exascale SW Project
- Big Data / Extreme Computing
- Collaborative Books
- Various National Reports

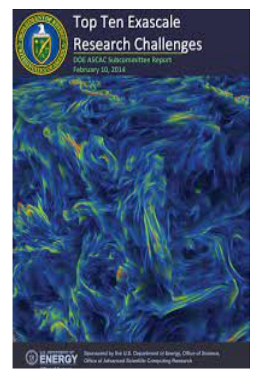


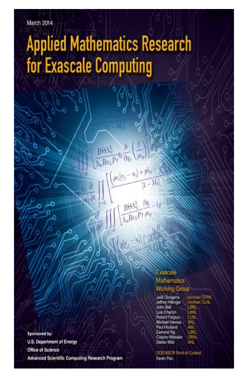












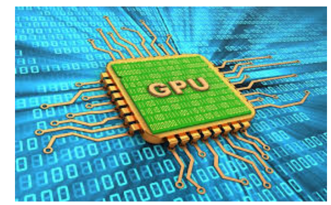
Great Group of Co

Today’s HPC Environment for Numerical Libraries

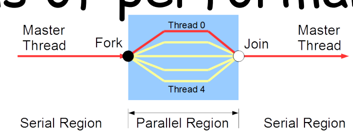
- Highly parallel
  - Distributed memory
  - MPI + Open-MP programming model

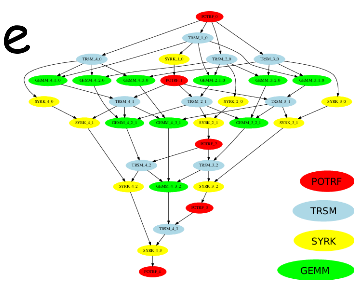
ORNL Summit, 200 Pflop/s, 4608 nodes (node= 2-Power9 chips + 6-Nvidia GPUs) 2.3 x 10^6 Cores

- Heterogeneous
  - Commodity processors + GPU accelerators

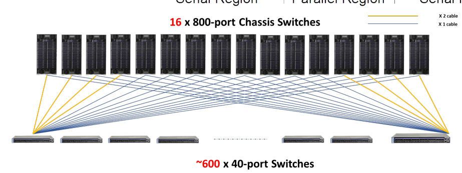


- Simple loop level parallelism too limiting in terms of performance





- Communication between parts very expensive compared to floating point ops



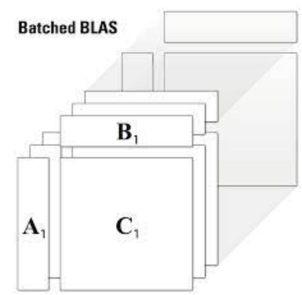
- Comparison of operation counts may not reflect time to solution

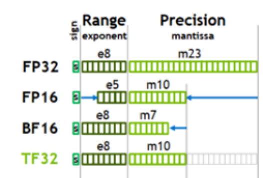
- Floating point hardware at 64, 32, and 16 bit levels

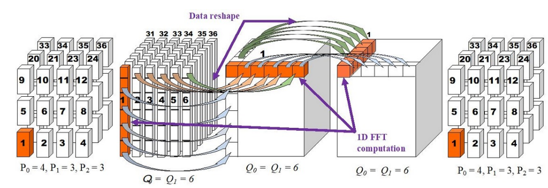
Type	Size	Range	$u = 2^{-t}$
half	16 bits	$10^{\pm 5}$	$2^{-11} \approx 4.9 \times 10^{-4}$
single	32 bits	$10^{\pm 38}$	$2^{-24} \approx 6.0 \times 10^{-8}$
double	64 bits	$10^{\pm 308}$	$2^{-53} \approx 1.1 \times 10^{-16}$
quadruple	128 bits	$10^{\pm 4932}$	$2^{-113} \approx 9.6 \times 10^{-35}$

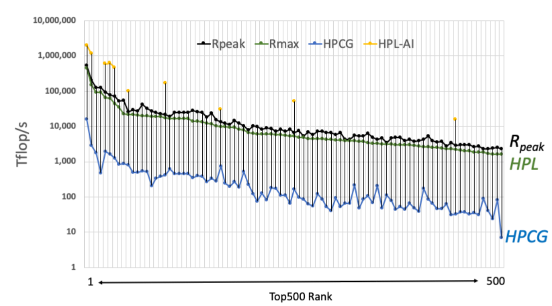
Today The Fun Continues at ICL …

- SLATE, PLASMA, MAGMA, BALLISTIC
- Batched BLAS
- Mixed Precision Computations
- heFFTe
- PAPI
- PaRSEC
- TOP500/HPL, HPCG, HPL-AI
  - My hobby







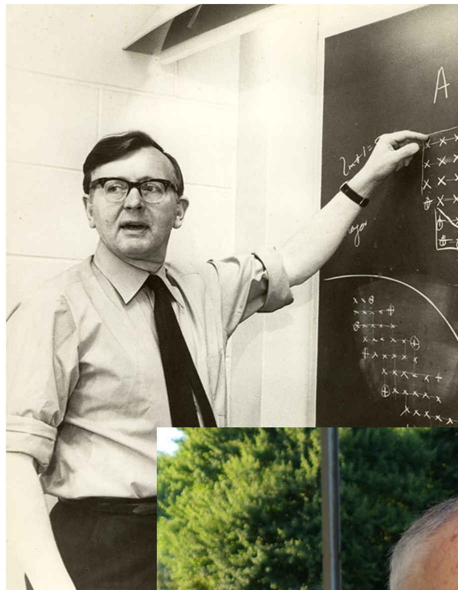


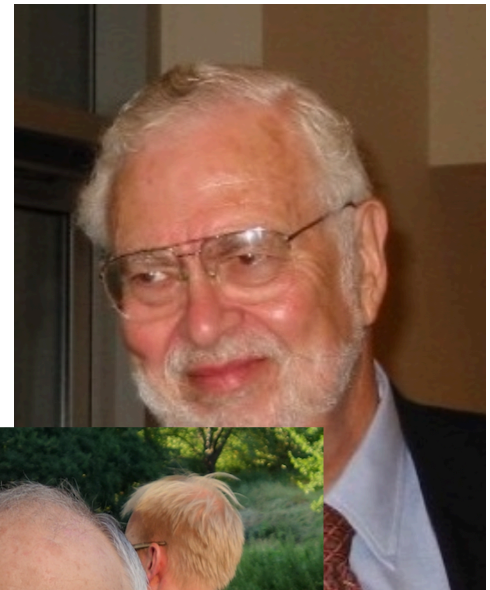
Innovative Computing Laboratory, 2019

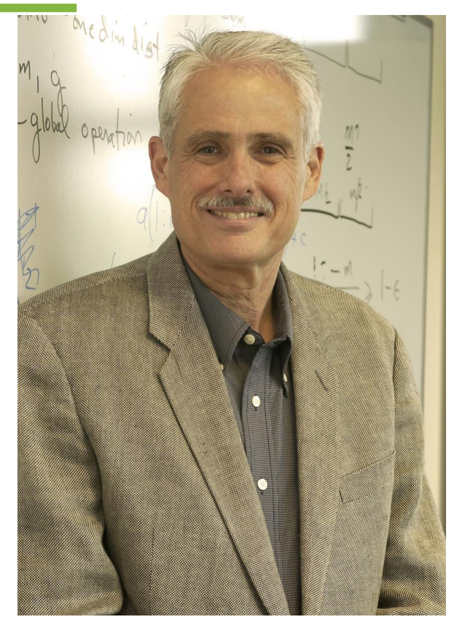
Stats from ICL: 33 PhD Graduates; 34 PostDocs; 135 Grad Students; 168 Staff

2019

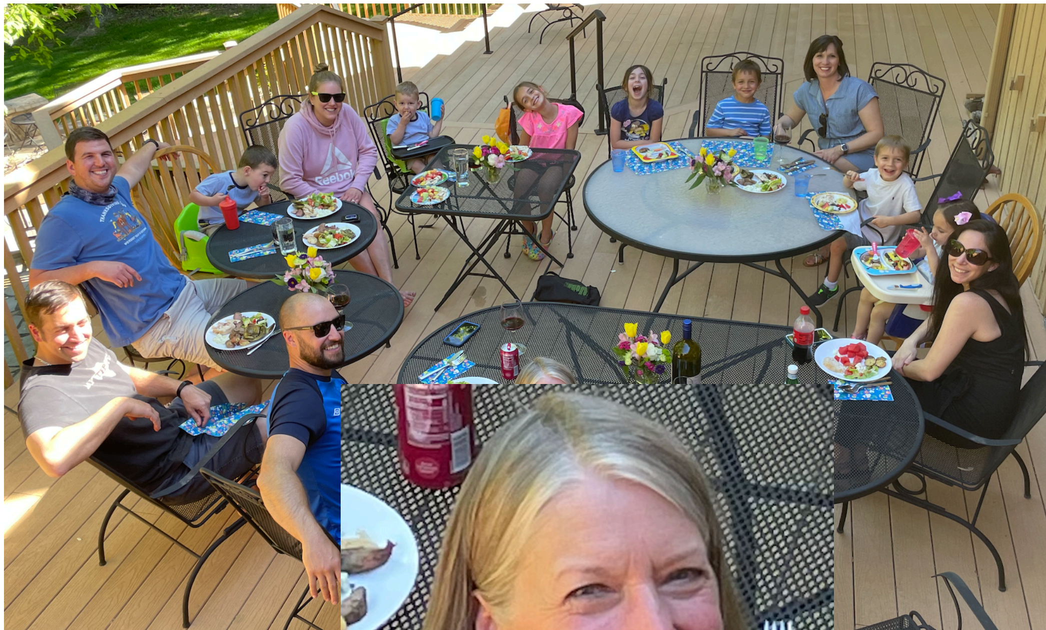
Mentors







Family



Jack’s 50 Years

Thanks to So Many People: Mentors, Colleagues Friends, Staff, and Students

Abdelfattah A	Agullo E	Anderson E	Angskun T	Anzt H	Arnold D
Ayala A	Baboulin M	Barrett R	Beck M	Beckman P	Becker D
Beguelin A	Berman F	Blackford S	Blanchard P	Bland W	Bosilca G
Bouteiller A	Boyle J	Browne S	Buttari A	Canning A	Cao C
Cappello F	Casanova H	Chan T	Charara A	Chen Z	Choi J
Chow E	Cowell W	Danalis A	Davis T	Demmel J	Desprez F
Dong T	Dongarra S	Du P	Ducroz J	Duff I	Eijkhout V
Fagg G	Faverge M	Finney J	Ford B	Fox G	Gabriel E
Garbow B	Gates M	Geist A	Golub G	Graham R	Grigori L
Gropp B	Guermouche A	Haidar A	Hammarling S	Herault T	Herrmann J
Heuveline V	Higham N	Jagode H	Johnsson L	Kabir K	Kagstrom B
Kennedy K	Keyes D	Kranzlmuller D	Kurzak J	Langou J	Langou J
Lemarinier P	Ltaief H	Lumsdaine A	Lusk R	Luszczek P	Lyness, J
Ma, T	Manchek R	Marques O	Matsuoka S	Mawussi Z	Messina P
Meuer H	Mohr B	Moler C	Moore K	Moore S	Moore T
Mrozek D	Mucci P	Nath R	Petitet A	Pjesivac-Grbovic J	Plank J
Pool J	Pozo R	Rafferty T	Reed D	Robert Y	Roche K
Schreiber R	Schulthess T	Seymour K	Shi Z	Simon H	Smith, B
Snir M	Solca R	Song F	Sorensen D	Srikara P	Stewart P
Strohmaier E	Sunderam V	Tisseur F	Tomov S	Tourancheau B	Tsai M
Vadhiyar S	VandeGeign R	VanderVorst H	Walker D	Wang L	Wasniewski J
Whaley R	Wilkinson J	Wolf F	Wolski R	Yamazaki I	YarKhan A

The ICL gang