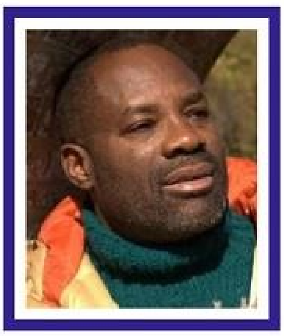
49 Father of Large-Scale Algebra—Part 2 of 10



Philip Emeagwali
info@emeagwali.com 202-203-8724

Philip Emeagwali Lecture 180613-1 and 170624
Visit http://emeagwali.com for complete transcripts of 100+ lectures.
Video: https://YouTube.com/emeagwali
Podcast: https://SoundCloud.com/emeagwali

49.1 Changing the Way We Do Large-Scale Algebra

49.1.1 Philip Emeagwali's Contributions to Large-Scale Algebra

I'm **Philip** **Emeagwali**.
I was in major U.S. newspapers,
such as the June 20, 1990 issue
of the *Wall Street Journal*
because I contributed
new calculus to modern calculus,
new algebra to extreme-scale algebra,
and invented a never-before-seen
computer
for parallel processing computer science.
I was the cover story
of top mathematics publications,
such as the May 1990 issue
of the *SIAM News*.
The *SIAM News*
is the most widely read mathematics
news journal.

I was the cover story
of top mathematics publications,
not because of my good looks
but because of my contributions
to mathematical knowledge, namely,
nine never-before-seen
partial differential equations
of modern calculus
that I invented.
Back in 1989, I was in the headlines
because I mathematically
and experimentally invented
how to solve
the most extreme-scaled problems
arising from the system of
partial difference equations
of modern algebra.
That system of
partial difference equations
approximates
the system of coupled, non-linear,

time-dependent, and state-of-the-art 2380
partial differential equations
of modern calculus.
That system of
partial differential equations,
in turn, encodes
a set of laws of physics
that governs the motions of fluids.
I was in the headlines
because I mathematically
and experimentally invented
how to massively parallel process
and how to communicate **synchronously**
and how to compute **simultaneously**
and how to do both **across**
my new internet
that is a new global network of
65,536 tightly-coupled processors
with each processor
operating its own operating system
and with each processor

having its own dedicated memory 2381
that shared nothing with each other
that is a new supercomputer
and that is a new computer.
I was in the headlines
because I mathematically
and experimentally invented
how to solve
the toughest problems arising
in the most extreme-scaled algebra
and how to solve them
on the most high-performance
massively parallel processing
supercomputer
and how to use that new knowledge
in extreme-scaled algebra
to look a mile deep into an oilfield
and help **recover**
otherwise **unrecoverable**
crude oil and natural gas.

My country of birth, Nigeria, 2382
benefitted immensely
from the technology
that is used to discover and recover
crude oil and natural gas.
The modern high-performance
supercomputer
that derives its power
from communicating
and computing
across millions upon millions
of tightly-coupled commodity processors
is the digital divining rod
that provides **visibility**
inside the **darkness**
of the mile-deep oilfields
of the Niger-Delta region
of southeastern Nigeria.
In modern times, the petroleum industry
purchases one in ten supercomputers
and purchased those supercomputers

2383

because the supercomputer
earns money, saves money,
and pays for its
100 million-dollar price tag.
The modern supercomputer
now draws its computing power
by supercomputing many things **at once**,
or in parallel, instead of computing
only one thing **at a time**,
or in sequence.
Back in 1989,
it made the news headlines
that a lone wolf
African supercomputer wizard
in Los Alamos, New Mexico,
United States
has invented
how to compute the solutions
of extreme-scale system of equations
of algebra
that arises from simulating

the motions of injected water, crude oil, 2384
and natural gas
that are flowing
across production oilfields.
I—Philip **Emeagwali**—
was that African supercomputer scientist
that was in the news
for inventing
how to solve
the largest system of equations
of modern algebra
and for inventing
how to solve them **at once**,
or in parallel,
instead of solving those equations
only one equation **at a time**,
or in sequence.
I was in the headlines
because I invented
how to solve 65,536 sets of equations
of modern algebra

and how to solve those equations **across** 2385
my new internet
that is a new global network of
65,536 tightly-coupled
commodity processors
that shared nothing with each other.
I was in the news because
my mathematical invention
changed the way we solved
the toughest problems
arising in extreme-scale algebra.
In the old way,
we solved many algebraic problems
and solved them one **at a time**.
We solved grand challenge problems
in sequence
and solved them within only one isolated
processor
that was not a member
of an ensemble of processors
that communicates and computes

together 2386
and as one seamless, cohesive
supercomputer.
In my new way
that made the news headlines
in 1989,
I invented how to solve
a set of 65,536
algebraic problems
that each comprised
of a system of 366 equations
of algebra
and each with 366 variables.
I invented
how to solve the most extreme-scaled
problems arising in algebra
and I invented
how to simultaneously solve
those problems and solve them **across**
those 65,536 processors,
or solve them in parallel.

Before my invention 2387
of the massively parallel processing
supercomputer
that occurred
on the Fourth of July 1989,
parallel processing
was controversial
and was widely dismissed
as a huge waste of everybody’s time.
Parallel processing, the technology
that powers
the modern computer
and massively powers
the high-performance supercomputer
rests on the intellectual confidence
that was gained
from that new knowledge
that arose from my invention
that occurred
on the Fourth of July 1989.
After my experimental invention

of 1989, 2388
the once controversial
parallel processing
became grounded in practical confidence,
or practical knowledge.
It's practical knowledge because
parallel processing is embodied
inside nearly every computer
and inside all supercomputers.
It's practical knowledge because
parallel processing
is at the foundation of
extreme-scale computational physics
as well as computational mathematics.
It's practical knowledge because
parallel processing
is used to **foresee** otherwise
unforeseeable climate change.
Global warming is a disaster-in-waiting
that is looming on the horizon
but that can be foreseen more accurately

with the massively parallel processing supercomputer
that I experimentally invented.

2389

49.1.2 Making the Impossible Possible

As an extreme-scaled
computational mathematician
that came of age
in the 1970s and '80s,
my grand challenge
was to show that
what's impossible-to-compute
for the computational physicist
—that is computing with the
fastest processor—
is, in fact, possible-to-compute
for the massively parallel processing
supercomputer scientist,

that is supercomputing 2390
and computing **across**
the slowest
65,536 processors in the world.
As a large-scale **algebraist**,
who is an expert in solving
the largest system of equations
arising in algebra,
I focused on the structure
and on the form
of my world record system of
24 million equations
of algebra
that was a world record in 1989.
In the petroleum reservoir
simulation codes
of the 1950s, ‘60s, and ‘70s,
the technique of choice
for discretizing the governing
partial differential equations
for crude oil and natural gas recovery

is called
the **alternating direction implicit**
finite difference method.
In modern textbooks
on finite difference discretizations
of partial differential equations,
the **alternating direction implicit**
method is also recommended
for solving the heat equation
that is the poster boy
of parabolic
partial differential equations.
The **alternating direction**
implicit method
yields a tri-diagonal
system of partial difference equations
of algebra,
or a system
in which its companion square matrix
has **nonzero** elements
only on its **diagonal**

row of entries
and along its **sub-diagonal**
row of entries
and along its **super-diagonal**
row of entries.
A tri-diagonal system of
partial difference equations
of algebra
is unsolveable in parallel.
Or to solve by processing many things
(or processes or equations)
at once.
A tri-diagonal system of
partial difference equations
is only possible to solve
in sequence.
Or to solve by solving
only one equation
at a time.
Sixty-five thousand
five hundred and thirty six [65,536]

subsets of
a tri-diagonal system of
partial difference equations
cannot be emailed **synchronously**
and/or solved **simultaneously**.
It's impossible to do both
across a new internet
that is a new global network of
65,536 tightly-coupled processors
that shared nothing with each other.
My contribution
to computational mathematics
was that I invented
how to reformulate
the tri-diagonal system of
partial **difference** equations
of extreme-scale algebra
that is used in the petroleum industry
and used to achieve
the highest resolution
in the petroleum reservoir simulators

that are used to **recover**
otherwise **unrecoverable**
crude oil and natural gas.
The tri-diagonal system of
partial **difference** equations
of extreme-scale algebra
arose from implicit
finite difference discretizations
and approximations
of the governing system of
partial **differential** equations
of modern calculus.
Implicit finite difference approximations
allow long time steps
which compress the amount of
floating-point arithmetical operations
that will be executed.
However, the tri-diagonal system
of partial **difference** equations
of extreme-scale algebra
cannot be solved in parallel.

On the other hand, 2395
the diagonal system of
partial **difference** equations
of extreme-scale algebra
obtained from explicit
finite difference discretizations
of the governing system of
partial **differential** equations
of modern calculus
only allow short time steps
which, in turn, increase
the amount of
floating-point arithmetical operations
that will be executed.
The short time steps
of explicit finite difference
discretizations
is determined by the Courant condition
that prescribes the relationship
between the length
of the spatial time steps,

the temporal time steps, 2396
and the wave speeds.
The computational fluid dynamics model
loses its mathematical accuracy
when its time-steps cannot account for
the small-scale factors
that affect the fluids flowing **across**
the surface of the Earth.
For these reasons, explicit
finite difference discretizations
are more computation-intensive
than implicit
finite difference discretizations.
In other words, short time steps
means more time steps
to keep the system of
partial difference equations
of extreme-scale algebra
from falling apart,
or becoming unstable.
I preferred explicit finite difference

approximations
because their diagonal systems
of partial difference equations
of extreme-scale algebra
can be solved in parallel.
I **theoretically discovered**
how to reformulate
my system of
partial differential equations
of modern calculus
into a diagonal system of
partial difference equations
of extreme-scale algebra that
approximated them
and that enabled me
to solve 65,536
problems **at once**.

2397

49.2 The Magical Power of Algebra

49.2.1 My Discovery in Large-Scale Algebra

I was asked
to describe how extreme-scale algebra
is used in day to day life in Africa.
In my country of birth, Nigeria,
and in any oil producing nation,
the largest possible systems
of partial difference equations
of extreme-scale algebra
must be solved
as a precondition
to discovering otherwise undiscoverable
crude oil and natural gas.
The dense, abstract, and invisible
large systems of
partial **difference** equations
of extreme-scale algebra
are the **common denominators**
across all supercomputers

that were **ever built**. 2399
And one in ten supercomputers
ever built
were purchased
by the petroleum industry.
The high-performance supercomputer
is used to discover and recover
otherwise elusive
and unrecoverable
crude oil and natural gas
from the Niger Delta oilfields
of the southeastern region of Nigeria.
The supercomputer
contributes to Nigeria's economy.
The larger the system of
partial **difference** equations
of extreme-scale algebra
that computational mathematicians
can solve
the more oil money
they can contribute
to build schools and hospitals
in Nigeria.

2400

The fundamental question
at the crossroad of the frontier
of the largest system of equations
of extreme-scale algebra
and the frontier
of the most massively parallel
processing supercomputer
is this:
Can a network of
eight processors
—that is the heartbeat
of the modern computer—
be harnessed
and used to solve
the largest system of
partial **difference** equations
of extreme-scale algebra?
Large systems of
partial **difference** equations
of extreme-scale algebra
arise from our societal needs
to make our world a better place,
and a more knowledgeable one.

Large systems of
partial **difference** equations
of algebra
arise from general circulation modeling.
And general circulation models
are used to forecast
and to foresee otherwise unforeseeable
global warming?
Large systems of
partial **difference** equations
of algebra
arise from our need
to recover otherwise unrecoverable
crude oil and natural gas
and to recover them
by using computational physics codes
and using those codes
to hindcast the motions
of crude oil and natural gas.

49.2.2 Algebra Enables Oil Recovery

The **Oloibiri oilfield** 2402
of Bayelsa State
of southeastern Nigeria
is the first oilfield in West Africa.
The **Oloibiri oilfield**
was abandoned in 1978.
Like all abandoned oilfields,
the **Oloibiri oilfield**
was abandoned merely twenty years
after it was discovered.
Like other dried up oilfields,
the **Oloibiri oilfield**
was not dried up in a literal sense.
About **70 percent**
of the crude oil discovered in **Oloibiri**
remains unrecoverable
and remains in **Oloibiri**.
That **70 percent** was abandoned
because that **70 percent**
will cost **71 percent** to recover.
I was asked:
"How do we recover crude oil
and natural gas?"

Briefly, to increase the crude oil
and natural gas recovered
from the Niger Delta oilfields
of the southeastern region of Nigeria,
water is pumped into each oilfield.
Massively parallel processing supercomputer
simulations
are used in advance
and used in Nigeria
and used to help petroleum engineers
that depend on
the unreasonable preciseness
of the laws of motion of physics.
To the extreme-scaled
computational physicist
simulating **across**
millions upon millions
of tightly-coupled commodity processors,
the set of laws of physics
is the **magic sword,**
or the hero’s sword.
That **magic sword** is not physical.
That **magic sword** is intellectual.

The massively parallel processing
supercomputer
is the **diving rod**
of the petroleum geologist.
That massively parallel processing
supercomputer rod
is imbued with the magical power
of a system of
partial **difference** equations
of extreme-scale algebra
that was formulated
from a companion system of
partial **differential** equations
of modern calculus
that embodied
a set of laws of physics.
Large-scale partial **difference** equations of
extreme-scale algebra
solved at the fastest
massively parallel processed
supercomputer speeds
is the new mathematical knowledge
that is used to understand

the best ways 2405
to pump water into the oilfields
of the Niger Delta Region
of southeastern Nigeria.
Pumping water
into production oilfields
drives the most crude oil and natural gas
towards crude oil and natural gas
production wells.
Back in the 1980s and earlier,
the 25,000 vector processing
supercomputer scientists in the world
that had **Seymour Cray**
as their spokesman
were demanding
an experiment-verified proof
that they could massively
parallel process **across**
millions upon millions
of tightly-coupled commodity processors.
Seymour Cray
and those 25,000 vector processing
supercomputer scientists

were not impressed 2406
by big ideas about
the massively parallel processing
supercomputer,
or deep theories on how to solve
the toughest
initial-boundary value problems
and how to solve them
in theory but not in practice.
In 1989,
it made the news headlines
that a lone wolf
African supercomputer wizard
in Los Alamos, New Mexico,
United States
had invented the massively parallel
processing
supercomputer.
I am that supercomputer scientist
that invented
the parallel processing technology
and invented it
as a new internet

and as a new global network of
65,536 tightly-coupled
commodity processors
that shared nothing with each other
and that can be used **to solve**
in one day
the most computation-intensive problems
arising in physics
and solve in one day
one of the toughest problems
arising in computational physics
and solve in one day
what formerly took
65,536 days,
or 180 years, of time-to-solution.
I—**Philip Emeagwali**—was that
African supercomputer scientist
that invented
how to massively parallel process
and massively compress
180 years of time-to-solution
to only one day of **time-to-solution.**
I invented

the Philip Emeagwali formula
that then U.S. President Bill Clinton
reconfirmed
in his White House speech
of August 26, 2000.

Before my invention, parallel processing
solved the toughest problems
in theory, but not in practice.
Before my invention,
it was commonly said that
parallel processing is good in theory,
but not in practice.
I immortalized parallel processing
in the computer.
Solving the toughest problems
arising in mathematics and physics
and solving them
across a new internet
do not work out the same way
for solving same problem
on a computer.
The outcomes of theorized
massively parallel processing

are different
when putting theory into practice.
A supercomputer theory
should be a subset of reality.

49.2.3 My Quest for the Largest-Scaled Algebra

I began my quest
for the largest system of equations
of algebra that can be solved
on June 20, 1974.
I began my quest
at the frontier of large-scale algebra
and I began that quest
by solving
the largest systems of equations
of algebra
that I could solve
on the fastest supercomputer
in the world.
I began my quest
with a tri-diagonal matrix algorithm

for finite **difference** equations 2410
of extreme-scale algebra.
That tri-diagonal matrix algorithm
is a simplified
Gaussian elimination algorithm
that is described in algebra textbooks.
That tri-diagonal matrix algorithm
is two orders of magnitude
more efficient
than the Gaussian elimination algorithm
applied to **non-sparse**
and **non-structured** matrices,
or applied to **dense** matrices
that had entirely **non-zero** elements.
That tri-diagonal matrix algorithm
is valid only for
diagonally dominant matrices.
I began my quest
in extreme-scale
computational mathematics
and **I began that quest**
by solving extreme-scale
systems of partial **difference** equations

of algebra 2411
and solving those equation
on a sequential processing supercomputer.
That supercomputer
was at 1800 SW Campus Way,
Corvallis, Oregon, United States.
In the mid-1970s,
and at 7:00 in the morning,
I parked my red two-speed bicycle
in the bicycle rack
that was behind Kidder Hall
at 2000 SW Campus Way.
I parked 200 feet away
from the sequential processing
supercomputer
that I was programming.
I had been supercomputing
since June 20, 1974,
and since age nineteen,
and, therefore,
I should not be described
as an overnight success.

49.3 Large-Scale Computations in Algebra

49.3.1 Opening the Door to Large-Scale Algebra

On the Fourth of July 1989,
I invented
the massively parallel processing
supercomputer.
I invented that technology
after fifteen years
of supercomputing alone.
My invention of how to solve
a then world record system
of 24 million
partial **difference** equations
of extreme-scale algebra
made the news headlines
because it was an invention
that opened the door
to extreme-scale algebra.
I invented

the massively parallel processing
supercomputer
and I invented the technology
only after the community of 25,000
supercomputer scientists,
that were led by **Seymour Cray,**
had given up
on the massively parallel processing
supercomputer.
I invented
the massively parallel processing
supercomputer
and I invented the technology
as heralding
the end of vector processing
supercomputers.
My invention
was a paradigm shift
because it forced 25,000
supercomputer programmers
and their leader, **Seymour Cray**,
to abandon
their vector processing supercomputers

and embrace
the modern supercomputer
that is powered by
parallel processing technology.

I'm **Philip Emeagwali**.

I discovered
the massively parallel processing
supercomputer
as the starting point
for the mass production
and the commercialization
of parallel processing computers
and massively parallel processing
supercomputers.
Parallel processing
enables the modern supercomputer
to do more than ten million things
at once.
Unlike the 25,000
vector processing
supercomputer programmers

of the decades of the 1970s and ‘80s 2415
and the sequential processing
supercomputer programmers
of the decades of the 1950s and ‘60s,
I ignored the widespread skepticism
that parallel processing
is a huge waste of everybody’s time.
I ignored the warnings
of the leaders of thought
that was published
in the June 14, 1976 issue
of the *Computer World*.
The *Computer World*
was the intellectual mouthpiece
of the computing community.
That issue of the *Computer World*
carried an article titled:

[quote]
“Research in Parallel Processing
Questioned as ‘Waste of Time.’”
[unquote]

49.3.2 Changing the Way We Do Extreme-Scale Algebra

The sequential processing
supercomputers
that I programmed
back in 1974
only executed
floating-point arithmetical operations
and executed them
on **pairs of numbers**.
A sequential processing supercomputer
computes in sequence,
or by computing
only one **pair of numbers**
at a time.
For my fast sequential calculations
of the mid-1970s,
I used the algebra textbook technique

called **Gaussian Elimination**.
That technique
is described in textbooks
on linear algebra.
I used a variant
of **Gaussian Elimination**
that required **no pivoting**.
I used that variant
of that classic algebraic technique
to solve my system of
tri-diagonal, or three-diagonal,
equations of algebra.
My **tri-diagonal** equations
arose from
the computational physics codes
that I wrote
for sequential processing
supercomputers
that I was programming
in the 1970s.
I wrote those codes

for **hindcasting**
the motions of crude oil, injected water,
and natural gas
that flows through a porous medium.
This includes the motions
of crude oil and natural gas
that flow towards production oil wells.
An oilfield or a water aquifer
is defined within a porous medium.
Put differently, the oilfield
is comprised of voids called “pores”
that are filled with crude oil
and natural gas
that flow **across** the voids
and flow **towards**
the production oil wells.
If this extreme-scaled
petroleum reservoir simulation
technique
was executed **across**
a massively parallel processing

supercomputer
was applied
back in 1958
and applied to recovering crude oil
and natural gas
from **Oloibiri Oil Field**
in Bayelsa State (**Nigeria**)
that oilfield
that was the first oilfield
that was discovered in West Africa
would not have been abandoned
in 1978,
or abandoned
merely twenty years
after it was discovered.

49.3.3 Contributions of Philip Emeagwali to Algebra

I'm **Philip Emeagwali**.
I am the mathematician and the physicist
that invented

how to solve the toughest problems 2420
arising in modern calculus,
extreme-scale algebra,
and computational physics.
On the Fourth of July 1989
and in Los Alamos, New Mexico,
United States,
I invented the precursor
to the massively parallel processing
supercomputer
of today.
My new supercomputer
is the technology that enables
the most accurate simulations
demanded for extreme-scaled
computational physics
within a multi-disciplinary environment.
I am well known
as the supercomputer scientist
that contributed to the development
of the modern computer.

2421

But I am not known well
as the mathematician
that contributed new equations
to modern calculus
and extreme-scale algebra.
I’m well known
for my invention
of the massively parallel processing
supercomputer.
But I’m not known well
for how my invention
changed the way
we look at the modern computer.
I’m well known
for my invention
of the high-performance supercomputer.
But I’m not known well
for the rich and fertile consequences
that my invention brought
to the supercomputer industry.

I am well known
but I am not known well.

2422