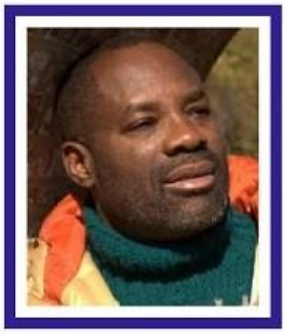
50 Father of Extreme-Scale Algebra—Part 2 of 10



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

Philip Emeagwali Lecture 180613-2 and 170624 Part 2
Visit http://emeagwali.com for complete transcripts of 100+ lectures.
Video: https://YouTube.com/emeagwali
Podcast: https://SoundCloud.com/emeagwali

50.1 Changing the Way We Do Large-Scale Algebra

50.1.1 Contributions of Philip Emeagwali to Algebra

I'm **Philip Emeagwali**.
I'm an extreme-scaled
computational algebraist.
Extreme-scale algebra
is the recurring decimal
in the discovery and recovery
of otherwise elusive
and unrecoverable
crude oil and natural gas.
The massively parallel processing
supercomputer
is the bedrock technology
that must be used
to discover and recover
crude oil and natural gas.
My invention
of the massively parallel processing
supercomputer
that occurred
on the Fourth of July 1989
that was recorded

in major U.S. newspapers
and recorded
in the June 20, 1990 issue
of the *Wall Street Journal*
opened the path
to excruciatingly-detailed
computational physics,
such as extreme-scaled
petroleum reservoir simulations.
My invention
of the massively parallel processing
supercomputer
changed the way we solve
the toughest algebraic problems
arising in extreme-scaled
computational physics.
My invention
of the massively parallel processing
supercomputer
inspired a new way
of high-performance supercomputing.

My invention
of the massively parallel processing supercomputer
enabled
computational mathematical physicists
to compress their time-to-solution
from a theorized 30,000 years
to just one day.
My invention
of the massively parallel processing supercomputer
led to a paradigm shift
in high-performance supercomputing.
In the old paradigm,
the high-performance supercomputer
was computing
sequentially, or solving only one
algebraic problem at **a time**.
In my new paradigm,
the high-performance supercomputer
is supercomputing

in parallel, or solving a million
algebraic grand challenge problems
at once.

50.1.2 Algebra on Vector Supercomputers

The fastest algebraic computations
of the early 1980s
were executed
on vector processing supercomputers.
Those vector processing supercomputers
operated on **pairs of lists of numbers.**
Executing floating-point
arithmetical operations
on **pairs of lists of numbers**
is an order of magnitude faster
than merely operating
on **pairs of numbers**.
That ability
to execute floating-point operations
on **pairs of lists of numbers**

inspired the invention
of the **cyclic reduction** algorithm.
That algorithm can be used
to solve a system of
tri-diagonal equations
of algebra
and solve them
with shorter **times-to-solution**.
The **cyclic reduction** algorithm
is slower
on sequential processing
supercomputers.
The **cyclic reduction** algorithm
is faster
on vector processing supercomputers.
The reason the **cyclic reduction**
algorithm is faster
on vector processing supercomputers
is that it takes advantage
of some parallelism
introduced by the

cyclic reduction algorithm
and introduced into the set of
floating-point arithmetical operations.
For the **cyclic reduction** algorithm
to work, the matrix must be
diagonally dominant.
That matrix
is the array of the coefficients
of the system of
finite **difference** equations
of extreme-scale algebra
that I formulated
and that arose from my discretizations
of the governing system of
partial **differential** equations
of modern calculus
that I invented
that is called
the **Philip Emeagwali** Equations
that arose from my encoding
of a set of laws of physics

that governs the motions
of the crude oil, injected water,
and natural gas
that is either simulated
across an isolated processor
or **across** an ensemble of processors.
The matrix is **diagonally dominant**
when the diagonal entry
of the matrix
is at least equal to
the sum of the two **off-diagonal** entries
of the same matrix.
The **cyclic reduction** algorithm
requires the execution
of more floating-point
arithmetical operations.
The trade-off is that
some of those floating-point
arithmetical operations
can be speeded-up
and operated as

pairs of lists of numbers.
One reason the technology
of parallel processing
was scorned, ridiculed, and rejected
by **Seymour Cray** and **Gene Amdahl**
and ridiculed
by the community
of 25,000 supercomputer scientists
of the 1980s
and dismissed by everybody
as a huge waste of everybody's time
was that
it was truly a waste of time
to attempt to solve
a system of equations
arising in extreme-scale algebra
that possessed **non-zero**,
non-diagonal entries
and, in particular,
to attempt to solve
those algebraic equations

in parallel.
It was truly a waste of time
to attempt to solve or solve those
algebraic equations
by processing many things (or processes)
at once.
However, for excruciatingly-detailed
crude oil and natural gas
reservoir simulations,
I invented how to reformulate
those algebraic equations
and how to reformulate them
from first principles
and reformulate them
by re-examining
the foundational calculus
and the foundational physics
from which those algebraic equations
arose.
In a nutshell, the 25,000
supercomputer scientists

of the 1980s,
demanded lockdown evidence
that they could
massively parallel process **across**
a new internet
that is a global network of
65,536 tightly-coupled
commodity processors
that shared nothing with each other.
I—**Philip Emeagwali**—won accolades
from supercomputer scientists
because I provided that
lockdown evidence
and I provided that evidence
on the Fourth of July 1989
in Los Alamos, New Mexico,
United States.

50.2 How I Opened Doors to Large-Scale Algebra

50.2.1 Contributions of Philip Emeagwali to Algebra

Over my first sixteen years
of supercomputing,
that is onward of June 20, 1974,
I gained practical
supercomputer knowledge,
namely, the deeper understanding
that the toughest
initial-boundary value problems
arising in extreme-scale
computational physics
were not defined by the system of
partial **difference** equations
of extreme-scale algebra
that govern those problems.
The toughest problem
in computational physics

was defined
by the laws of physics
from which the
initial-boundary value problem
of calculus
that governs say,
a general circulation model, arose
and from which the
system of partial **difference** equations
of extreme-scale algebra
that governs say,
a petroleum reservoir simulation,
also arose.
I theoretically discovered
that I could go back
to the root of the
toughest problem
in extreme-scale computational physics.
That root
is some set of laws of physics.
From that root or first principles

or laws of physics,
I reformulated
the grand challenge
initial-boundary value problem
of extreme-scale computational physics—
called petroleum reservoir simulation—
and I reformulated that problem
to derive its governing system of
partial differential equations.
The **paradigm shift**
is that my new system
of partial differential equations
is **hyperbolic.**
The old system
of partial differential equations
that was taught
to mathematical geophysicists
and that was written
in textbooks
on porous media flows
is **parabolic**.

That paradigm shift or change
from parabolic
partial **differential** equations
to hyperbolic
partial **differential** equations
changed the Courant Condition
on the explicit
finite **difference** algorithms
that I invented
and that gave rise to my companion
partial **difference** approximations.
My mathematical discovery
of the early 1980s
in Washington, D.C., United States,
of that order of magnitude change
in the Courant Condition
allowed me to massively parallel process
and to massively compress
the **time-to-solution**
and to compress that time
by an order of magnitude.

I'm **Philip Emeagwali.**
I'm well known
as a supercomputer scientist
that invented a new supercomputer.
But I'm not known well
as a research mathematician
that invented
nine partial **differential** equations
of modern calculus.
I was the cover story
of top publications in mathematics
not because of my good looks.
I was cover stories
because I contributed to mathematics.
One of my contributions to mathematics
was that I invented
how to discretize my new system
of partial **differential** equations
of modern calculus
so that I can obtain
a new and approximating system of

partial **difference** equations
of extreme-scale algebra.
My reformulated algebraic problem
is **diagonal**
and, with that new **diagonal** structure,
my initial-boundary value problem
could be solved in parallel,
or solved **across** a new internet
that is a new global network of
two-raised-to-power-sixteen
commonly available microprocessors.
To put my contributions
to extreme-scale
computational mathematics
in a better perspective,
it should be noted that
the number of mathematical papers
that could be written is **infinite**.
But the number of
mathematical algorithms
that can be invented

is **finite.**
My contributions
to the partial **differential** equations
of modern calculus
and to solving the extreme-scale
algebraic problems
that arise in computational physics
and to solving
the extreme-scale systems of
partial **difference** equations
of algebra
and to solving those problems **across**
a new internet
that is a new global network of
processors
that were already available in the market
was one of the few
mathematical discoveries
that were cover stories
for mathematics publications.
The cover story

of the May 1990 issue
of the *SIAM News*,
the top mathematics news journal,
was on how I—**Philip Emeagwali**—
invented how to solve 65,536
initial-boundary value problems
of modern calculus
and invented how to solve them
at once,
instead of solving them
one
initial-boundary value problem
at a time.

50.2.2 A Mathematics Olympics, Revisited

My cover story
of the May 1990 issue
of the *SIAM News*,
the most widely-read

mathematics news journal,
lead to invitations
for me to speak face-to-face
and discuss my contributions
to mathematics.
I was invited to discuss
my contributions with the foremost
computational mathematicians.
On July 8 and 9, 1991
and in my adopted hometown
of Washington, D.C., United States,
I presented the mathematical details
of my contributions
to the most extreme-scaled
partial **difference** equations
of algebra.
I presented the computational details
of my contributions
to solving those problems in parallel,
or how I solved
65,536 algebraic problems

at once.
My contributions
to the most extreme-scaled
partial **difference** equations
of algebra
arose from my contributions
to partial **differential** equations
of modern calculus.
On July 8 and 9, 1991,
I formally presented
my contributions to mathematics
and I presented the nine
Philip Emeagwali
partial **differential** equations
of modern calculus
that I invented
and I presented the new system of
partial **difference** equations
of extreme-scale algebra
that I formulated
and I presented

both systems of equations
before a **standing-room** only audience
of research mathematicians.
My formal presentation
of new mathematics
to research mathematicians
took place at 11:00 in the morning
in the Dover Room
of the Washington Sheraton Hotel,
Washington, District of Columbia,
United States.
I formally presented my contributions
of new mathematics
to **standing room** only audience
of research mathematicians
that were attending
the International Congress
of Industrial and Applied Mathematics.
At the end of my lecture
on my new calculus
from which arose

the largest system of
partial difference equations
of algebra
ever solved,
all the research mathematicians
seating in the **standing-room** only
Dover Room
of the Washington Sheraton Hotel
stood up
and gave me a **standing ovation**.
Like the Olympic games,
that conference holds once
every four years
and holds on different continents.
That conference is the Olympics games
for research mathematicians.
The speakers at **that conference**
were research mathematicians
that made **contributions**
to mathematics.
The speakers

at **that mathematics conference**
were like the gold medalists
in research mathematics.
As a speaker
at **that mathematics conference**,
I was like a gold medalist
in algebra
that was recognized
for inventing new algebra
and a gold medalist
in calculus
that was recognized
for inventing new calculus.
At **that mathematics conference**,
I spoke in the arcane language
of research mathematicians.
I used mathematical short hands
that enabled me to describe
my **contributions** to calculus
and to algebra
and to describe

my mathematical contributions
with only a few
never-before-seen
partial **differential** equations
of modern calculus.

50.3 How I Pushed the Frontier of Algebra

50.3.1 Contributions of Philip Emeagwali to Algebra

After describing my new
system of nine coupled, non-linear,
time-dependent, and state-of-the-art
partial **differential** equations
of modern calculus
that were hyperbolic,
instead of parabolic,
I explained that:
my new system of

24 million diagonal
partial **difference** equations
of extreme-scale algebra
that was a world record
in 1989
that was the toughest problem
in extreme-scale algebra
that arose from
explicit finite **difference** approximations
of **hyperbolic**
partial **differential** equations
was not equivalent
to the old system
of **tri-diagonal** equations
of algebra
that also arose from
implicit finite **difference**
approximations
of **parabolic**
partial **differential** equations
that govern the motions

of crude oil, injected water,
and natural gas
that flow **across** and a mile-deep
inside a porous medium,
such as the Niger-Delta oilfields
of southeastern Nigeria.
At that international congress
for research mathematicians,
I explained that
the polymath
—that is a triple threat
at the frontiers of knowledge
in physics, mathematics,
and supercomputing—
looks beyond his blackboard,
looks towards his motherboard,
and looks **across**
a new internet
that is a new global network of
64 binary thousand motherboards.
Unlike the research mathematician

who only looks at his blackboard,
the polymath looks **across**
a new internet
and did so to **theoretically**
and experimentally discover
that a system of diagonal equations
of algebra
that arose from solving
a specific grand challenge problem
in extreme-scale computational physics
need not be equivalent
to the system of **tri-diagonal** equations
of algebra
that also arose
from solving the same specific problem
in extreme-scale computational physics.
However, they both must solve
equivalent problems
in extreme-scale computational physics,
and solve those problems correctly.
They both must be equivalent

in the governing set of laws
of physics
from which each system of equations
of extreme-scale algebra
arose.

50.3.2 Opening Doors to Large-Scale Algebra

In 1989,
it made the news headlines
that an "African Supercomputer Wizard"
in Los Alamos, New Mexico,
United States,
had **invented** how to solve
the largest system of
partial **difference** equations
arising in algebra.
I—**Philip Emeagwali**—
was that African supercomputer scientist
that was in major U.S. newspapers

back in 1989 and thereafter.
I was in the June 20, 1990 issue
of the *Wall Street Journal*
because I invented
how to solve the toughest problems
arising in extreme-scale algebra.
I invented how to solve
initial-boundary value problems
of modern calculus
and how to solve those problems **across**
my new internet
that is a new global network of
64 binary thousand
commonly available processors.
Or a global network of
as many computers
that were identical
and that are equal distance **apart**.
It was across those
64 binary thousand
processors

that were already available in the market
that I connected
with the power of algebra.
I—**Philip Emeagwali**—was that
African supercomputer scientist
that was in the news in 1989
and thereafter.
I was in the news because
I experimentally invented
how to solve a world record system of
24 million
partial **difference** equations
of algebra
and I discovered
how to solve those algebraic problems
across a world record
number of commonly available
processors
and how to process them **in parallel**,
or by processing many things
(or processes)

at once.
And my invention opened the door
to extreme-scale algebra
that arises in extreme-scale
computational physics.
The system of equations
that I invented
were both **differential** and **algebraic**.
My **algebraic** equations
arose from my **differential** equations
that arose from the laws of physics.
Both systems of mathematical equations
could be used to **discover** and **recover**
otherwise elusive
and **unrecoverable**
crude oil and natural gas.

50.3.3 The Quest for Progress

It's been said that
they are more people living today
than ever lived before.
They are more scientists living today
than ever lived before.
A century ago,
they were only one thousand physicists
in the world.
Today, we have
one million physicists
in the world.
Three centuries and three decades ago,
Isaac Newton
did not describe himself as a scientist.
The word "**science**"
was unknown to **Isaac Newton**
who described himself
as a "**natural philosopher**."
Isaac Newton's knowledge of
mathematics
is at the 6th grade level

and **Isaac Newton**
lacks the mathematical maturity
needed to understand
the **Philip Emeagwali**'s equations
of modern calculus
that was the cover story
of the May 1990 issue
of *SIAM News*
that is the flagship publication
of the research mathematics community.
The **Isaac Newton** of the 17th century
lacked the knowledge
and the mathematical maturity
that is needed to understand
how to solve
an initial-boundary value problem
of modern calculus.
So our greater knowledge of our universe
is what sets us apart
from our scientific predecessors.
And the most important contribution

to our existing body of knowledge
is the scientific discovery
and the technological invention.
We should **distinguish**
between a discovery
and applying that discovery.
Or distinguish
between a discovery in physics
and applying that discovery
in engineering.
Or distinguish
between a discovery in biology
and applying that new knowledge
in medicine,
Or distinguish
between a scientific discovery
and a technological invention.
Using the modern supercomputer
to **discover** otherwise **undiscoverable**
crude oil and natural gas is merely
a geological discovery,

not a scientific discovery *per se*.
The discovery of crude oil
and natural gas
is an application of the invention
of the modern supercomputer
that computes in parallel.
For the research scientist
and technologist,
to not discover
and to not invent
is to not contribute to human knowledge
and is to slow down human progress.

50.4 My Fastest Computations in Algebra

50.4.1 Philip Emeagwali's Equations

After arithmetic,
calculus and algebra

are the most important branches
of mathematics.
Calculus
is the most powerful
mathematical technique of all times.
Yet few people understand
where **calculus**
comes from
and why and how **calculus** is even used
to foresee otherwise unforeseeable
global warming.
Or to deeply understand
how the **Philip Emeagwali** equations
can be discretized, coded,
and message-passed
to 65,536 processors
that shared nothing with each other
and used to discover and recover
otherwise elusive
and unrecoverable crude oil
and natural gas.

The system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern **calculus**
—that is also known as
Philip **Emeagwali**'s equations—
is used by first discretizing,
or approximating it,
to yield an extreme-scale system
of equations of algebra.
Extreme-scale algebra
arising from Philip **Emeagwali**'s
equations
demands extreme-scale computations
that, in turn, demands
a network of millions upon millions
of tightly-coupled processors
that shared nothing with each other
that **defined** and **outlined**
the fastest massively parallel processing
supercomputer

in the world.
Like new branches of mathematics,
my new calculus arose from
stimulations
from the world of petroleum reservoir
simulations,
or simulating the motions of
injected water, crude oil, and natural gas
that flows from water injection wells
towards crude oil production wells.
That **calculus** is at the bedrock
of such simulations
should not come as a surprise.
The reason is that in calculus
the **continuous** motion is decomposed
into **discrete** sequence of motions.
In 1989, my invention
of parallel processing
attracted international attention
and was in major U.S. newspapers.
My invention

of the massively parallel processing
supercomputer
that occurred at 8:15 on the morning
of Tuesday the Fourth of July 1989
and that occurred in Los Alamos,
New Mexico, United States,
made the news headlines
because that invention
was the first time EVER
the speed of computation
was **instantaneously**
and experimentally increased
and speeded up
by a factor as large as
65,536.
I attracted media attention
because my invention
of the massively parallel processing
supercomputer
yielded a measurable increase
in the speed of computation

that became my quantifiable contribution
to the development of the
modern supercomputer.
I spoke directly with the media
and I did so
to cut out the middleman scientist,
to cut out the opportunistic scientist,
that contributed nothing to my invention
but demands half of my invention credit
and demands half
of my invention earnings,
and demands them in exchange
for nothing
and in exchange
for helping me tell and share my story.
My nine partial differential equations
of modern calculus,
called the **Philip Emeagwali** equations,
and the companion
partial **difference** equations
of extreme-scale algebra

that approximated my nine equations
were not physical
never-before-seen objects
that I invented.
My new calculus
and my new algebra
were obstacles I overcame
in my quest for the fastest
massively parallel processing
supercomputer.
That new supercomputer
was a new internet
that was a new global network of
64 binary thousand tightly-coupled,
commonly available processors
that shared nothing with each other.
My new partial **differential** equations
and my new
partial **difference** algorithms
were my new research instruments
as much as my new processors

and email wires.
Sitting alone for a decade
and programming alone
the massively parallel processing
supercomputer
isn't child's play.
In the 1980s, I wrote **65,536**
supercomputer codes
that each solved
an initial-boundary value problem
that arose in extreme-scale
petroleum reservoir simulation.
I message-passed
those supercomputer codes
to my, two-raised-to-power-sixteen,
or **65,536,** commonly available
tightly-coupled processors
with each processor
operating its own operating system
and with each processor
having its own dedicated memory
that shared nothing with each other.

And I had as many sixteen-bit
addressed emails
that must traverse **across**
my sixteen times
two-raised-to-power-sixteen, **1,048,576,**
bi-directional email wires
that I visualized
as tightly encircling
the fifteen-dimensional **hypersurface**
of a **hypersphere**
in a sixteen-dimensional **hyperspace**.
In the mid-1980s,
my invention
was not recognized as an invention
in supercomputing.
It was only recognized as an invention
after I broke the speed records
in both computation and communication.
That invention
is **the Philip Emeagwali formula**
that then U.S. President Bill Clinton

reconfirmed
in his White House speech
of August 26, 2000.
In 1989, my invention
became a dramatic
and a concrete confirmation
that abstract equations
and invisible algorithms
could be simultaneously solved
across a **sixteen-network-deep** internet
that emulates
one cohesive supercomputer.
The new **equations** that I invented
and the new **algorithms** that I invented
represented my theory
while the processors
and email wires
represented my experiments.
Historically, the computer
was a "mathematics machine."
The new equations on the blackboard

and the new algorithms,
or the new equations
on the 64 binary thousand
motherboards,
were necessary and sufficient tools
for recording my arrivals
at the frontier of the new computer
and the new internet.
My goal was not to perform
the fastest computation
and fastest communication
per se.
My goal was not to solve my new
partial differential equations
and solve them to the 17th decimal place.
My goal was to see the plural
as the singular.
My goal was to see **65,536**
tightly-coupled, commonly available
processors
that shared nothing with each other

and to see that ensemble of processors
as one seamless, cohesive supercomputer
that is **65,536** times **faster**
than one computer
powered by a processor
that was not a member
of an ensemble of processors.
My goal was to see
a sixteen-network-deep internet
and see that new internet
as a small copy
of the Internet of the future
that could become
the new planetary-sized supercomputer
and become the new computer
for your children's children.