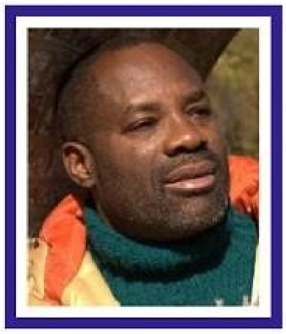
17 Father of Large-Scale Algebra



Philip Emeagwali Lecture 180605
Visit http://emeagwali.com for complete transcripts of 100+ lectures.
Video: https://YouTube.com/emeagwali
Podcast: https://SoundCloud.com/emeagwali

17.1 Father of Large-Scale Algebra

Please allow me to introduce myself.
I’m **Philip Emeagwali**.

17.1.1 The Grand Challenge Problem

I invented how to solve
the Grand Challenge Problem
of mathematics.
That Grand Challenge Problem
does not lend itself
to an analytical solution
that can be proved
to be correct.
In principle, the Grand Challenge Problem
of mathematics
could crudely be solved
on only one computer
that is powered by
only one processor.
Back in 1989, I made headlines in major U.S.
newspapers because I had invented
something new—namely,
the Philip Emeagwali formula
for solving
the Grand Challenge Problem
and for solving it accurately

and for solving it **across**
a new global network
of tightly-coupled processors.
That global network
that I invented
is a new internet
that is a new supercomputer
and that is a new computer.
Yet, I cannot completely define
in one hour
the Grand Challenge Problem
that the massively
parallel processing supercomputer solves.

17.1.2 Why Solve the Grand Challenge Problem

What I will say is that
the parallel processing
supercomputer technology
that I invented
on the Fourth of July of 1989
changed the way
we look at the modern computer.

Parallel processing
is a computer invention
that has rich and fertile consequences.
The parallel processing computer
is an invention
that makes the world
a more knowledgeable place
and a better place
for human beings
and for all beings.

17.1.3 Who is Philip Emeagwali?

I was a research supercomputer scientist
of the nineteen seventies and eighties
[1970s and '80s].
In those two decades, I conducted
undisturbed, unprecedented,
and bleeding-edge
supercomputer research
and speed measurements
and I conducted both
at the farthest frontiers

of the fastest computations ever recorded. 526
I conducted experiments
on how and why
the speed of the modern supercomputer
increases
when supercomputing **across**
a new internet.
I visualized and invented
that new internet
as a new ensemble,
or as a new global network
of 65,536 tightly-coupled
commodity-off-the-shelf processors
that were married together
by 1,048,576
email wires.
I invented that new internet
as a new supercomputer
and as a new computer.
That new supercomputer
is best used to solve
the toughest problems
arising in extreme-scale

algebraic computations
that, in turn, arose from
the partial differential equations
of calculus.
Those partial differential equations, in turn,
arose from the laws of physics,
such as the laws of
conservation of mass,
conservation of momentum,
and conservation of energy.
In supercomputing,
the light at the end of the tunnel
becomes brighter
with each supercomputer speed increase
that is experimentally confirmed.
The reason a mathematician
from my country of birth, Nigeria,
developed cold feet
over parallel processing
the most extreme-scaled problems
arising in algebra
is that mathematician's
lack of the understanding

of how and why
the most extreme-scale problems
in algebra
arises from the partial differential equation
of modern calculus.
The reason Nigeria
did not fund my research
in the parallel processing supercomputer
and the reason the research
computational mathematician
in Nigeria
—of the 1970s and ‘80s—
shied away from physics-based simulations
that were extremely computation-intensive
modern calculus
and why that algebraic knowledge
must be used to discover and recover
otherwise the elusive
crude oil and natural gas
that is buried one mile deep
inside the oilfields
of the Niger Delta region
of southeastern Nigeria.

I was asked to explain
how my contribution to algebra
can be studied
in your high school algebra class.
In high school algebra,
you learned how to use pencil and paper
to exactly solve for three unknowns
from a coupled system
of three simultaneous equations.
Your high school algebra textbook
did not attempt to teach you
how to exactly solve for four unknowns
from a coupled system of four,
or more, simultaneous equations.
The reason your high school algebra teacher
did not cover
a system of four simultaneous equations
was that the arithmetical operations count
that must be executed
to know the four unknowns
was more than you can handle.
For that reason,
extreme-scale problems in algebra—such as,

solving for a million billion trillion
unknowns
and solving for them
from a coupled system
of a million billion trillion
simultaneous equations—must be solved
not on your computer
but solved **across** a new internet
that is a new supercomputer
and a new computer.
On the Fourth of July 1989,
I mathematically and experimentally
invented
how to **exactly solve** for
24 million unknowns
and solve them from a coupled system
of 24 million simultaneous equations
the was the largest algebraic problems
arising from any physics-based simulations
within a multi-disciplinary environment
of the decade of the 1980s.
My world record solution
of the toughest problem

arising in algebra
was the cover story of the top mathematics
publication, namely, the May 1990 issue
of the *SIAM News*.
I was the cover story
of top mathematics publications
not because I was good looking.
I was the cover story
because my mathematical invention
was a giant leap forward
in the development of the modern algebra.
I invented
how to discover and recover
otherwise elusive crude oil and natural gas.
Back in 1989,
the 24 million equations of algebra
that I invented how to solve
and invented how to solve them
simultaneously
and **across**
a new global network
of 65,536 tightly-coupled,
commodity-off-the-shelf processors

that shared nothing with each other—
that is a new internet
by definition—was unsolveable
on the most powerful
vector processing supercomputer.
A 12-year-old writing a school report titled:
"Famous Mathematicians
and their Contributions to Mathematics"
asked me:
"What is the contribution
of Philip Emeagwali to algebra?"
I answered that
my contribution to algebra
is this:
I mathematically and experimentally
invented
how to parallel process **across**
a new internet
that is outlined and defined
by 65,536 identical processors
that uniformly encircled a globe.
On the Fourth of July 1989,
I invented

how to parallel process
and how to push
the frontier of knowledge
of modern algebra
and how to push that frontier
by a breath taking factor of
65,536.
Shortly after my invention
that occurred on the Fourth of July 1989,
it made the news headlines
that I—**Philip Emeagwali**—had invented
how to communicate and compute
and how to execute both collectively
and **across**
a new internet
that I **defined**
as a new global network of
65,536 processors
and that I invented
how to solve the toughest problems
in extreme-scale algebra
and I invented
how to solve those problems

65,536 times faster
than one processor
solving the same problem alone.
That 65,536-fold increase
in supercomputer speed
increases the user's productivity
while reducing the user's time-to-market.
In the old way of solving
the most computation-intensive problems
in large-scale algebra
and before my invention,
the most extreme-scaled
algebraic computations
were solved **across**
a customized and expensive vector processor
that processed
only one thing, or string of numbers
called vectors, at a time.
That vector processing supercomputer
was the fastest supercomputer
in the decades of the 1970s and '80s.
But in my new way,
that is, the new

parallel processing supercomputer,
that I mathematically and experimentally
invented
on the Fourth of July 1989
in Los Alamos, New Mexico, United States,
I figured out
how to solve, in parallel,
the most extreme-scale problem
in algebra
and I invented
that supercomputer technology
by massively parallel processing
that algebraic problem **across**
65,536 commodity-off-the-shelf processors
that I visualized
as a new internet
that is a new global network of
two-raised-to-power sixteen
processors
that were identical
and that were equal distances
apart.

It was for a good reason that
massively parallel processing
the Grand Challenge Problem of mathematics
and supercomputing **across**
millions upon millions of processors
was rejected as impossible
and described as a grand challenge.
As a lone wolf
massively parallel processing
supercomputer scientist
of the 1970s and '80s,
that was not a member
of a 400-person supercomputing team,
there were times I felt like
I was like a solo air traffic controller
on planet Earth
that was tasked
to direct 65 thousand identical planes
in the air
and tasked
to simultaneously, safely,
and properly land those 65 thousand
identical planes

and land them on
65,000 identical runways
that were equal distances
apart
and on the surface of a globe
the size of the Earth.
That was the reason
the June 14, 1976 issue
of the *Computer World* magazine
scorned parallel processing
and mocked the then unproven technology
as a huge waste of everybody's time.
The toughest problems
in computational physics
are linked by a common thread, namely,
extreme-scale algebra
that, in turn,
must be massively parallel processed **across**
millions upon millions
of commodity-off-the-shelf processors.
In the old paradigm
of algebraic computation,
the most computation-intensive problems

in algebra
were sequentially solved
by executing **only one**
floating-point arithmetical computations
at a time.

17.1.4 Diary of an Extreme-Scale Algebraist

In the old paradigm
of algebraic computation,
it was mathematically impossible
to email subsets
of the tri-diagonal system of equations
of algebra
and email them **across**
my ensemble of 65,536
commodity-off-the-shelf processors.
In my new paradigm
of explicit finite difference
algebraic approximations,
I made the impossible
possible

and I did so
by re-imagining the toughest problem
arising in calculus, namely,
the excruciatingly-detailed simulations
within a multi-disciplinary environment
of three-dimensional, three-phase flows
of crude oil, injected water, and natural gas
that were flowing **across**
a production petroleum reservoir
and re-envisioning
my governing partial differential equations
as **hyperbolic**
and doing so
when the calculus textbooks
described those equations as **parabolic**.
My inventions
of those coupled system of
nine partial **differential** equations,
called Philip Emeagwali's Equations,
of a new calculus
and my inventions
of their companion and approximating
nine partial **difference** equations

of a new algebra
is my contributions
to mathematical knowledge.
My contributions to mathematics
made the news headlines
across major U.S. newspapers
and they were the cover stories
of the top mathematics publications,
such as the May 1990 issue
of the *SIAM News*.
The *SIAM News* was published by
the Society for Industrial
and Applied Mathematics.
It was not my looks, wealth,
or family connections
that put me on the cover
of top publications
in the field of mathematics.
It was the nine Philip Emeagwali's equations
of modern calculus
that I invented
that put me on the cover
of the *SIAM News*

and on two postage stamps.
My contributions
to computational mathematics
changed how the petroleum industry
uses the modern supercomputer
to discover and recover
otherwise elusive
crude oil and natural gas.
Before my invention,
or before the Fourth of July 1989,
petroleum reservoir simulations
were executed on only one processor
that was not a member
of an ensemble of processors.
After the Fourth of July 1989
and today, all extreme-scale
petroleum reservoir simulations
and computational physics codes
are executed **across**
an ensemble of up to
ten million
six hundred and forty-nine thousand
six hundred [10,649,600]

commodity-off-the-shelf processors.
The first oilfield in West Africa
was discovered in 1956
and at **Oloibiri**
in the southeastern region
of colonial Nigeria.
Because the oil discovered in 1956
at **Oloibiri** Oil Field of Nigeria
was buried one mile deep,
we cannot see its one-mile deep movements
towards the crude oil and natural gas
production wells
that were also one-mile deep.
We cannot see the one-mile deep
subterranean motions
of crude oil and natural gas
with our biological eyes.
However, we can use our mathematical eyes
to **abstract**
the three-dimensional,
three-phased subterranean motions
of crude oil, injected water, and natural gas
and to **hindcast**

how the most crude oil and natural gas
will flow towards production wells.
Because the crude oil and natural gas
discovered **offshore**
of the southeastern Nigerian coastal waters
and deep under the ocean floor
or discovered **onshore** in Oloibiri (Nigeria)
was one mile deep,
we cannot see how that crude oil
and natural gas flows
from water injection wells
to crude oil
and natural gas production wells.
We cannot directly see
the flowing oil and natural gas
and see the oil and natural gas directly
as we directly see water flowing **across**
the River Niger of West Africa.
But our geological data
from the Niger-Delta region
of southeastern Nigeria
that includes the initial
and boundary conditions

of the governing
initial-boundary value problem
and its associated system of
partial differential equations,
and our knowledge of the laws of physics
and my invention
of the massively parallel processing
supercomputer
that occurred
on the Fourth of July 1989,
enables us to see **across**
my new internet
that I invented
as a new global network of
65,536 tightly-coupled processors
that were common available
in the market
and that were equal distances
apart.
That is, the massively parallel processing
computational physicist
can intellectually see
within a massively parallel processing

supercomputer
and see crude oil and natural gas
that the vector processing
computational physicist
cannot intellectually see
and that the ordinary person
cannot even see
with his biological eyes.
The massively parallel processing
computational physicist
that mathematically sees
deep inside the Niger-Delta oilfields
of southeastern Nigeria
enables us to discover and recover
otherwise elusive
crude oil and natural gas.
In 1989, it made the news headlines
that I invented
how we can use our parallel processing
supercomputer eyes,
or use a new internet
that is a new global network of
tightly-coupled, commodity processors,

as our instrument of large-scale physics
as well as use the supercomputer technology
as our tool for crude oil and natural gas
exploration.
Conversely, if the petroleum industry
didn’t accept my invention
and didn’t harness
my ensemble of 65,536, or more,
commodity-off-the-shelf processors
and didn’t use them
in their petroleum reservoir simulations,
then less crude oil and natural gas
will be discovered and recovered.
My invention
of the massively parallel processing
supercomputer
changed the way
the petroleum industry **discover**
and **recover**
otherwise elusive
crude oil and natural gas.
My discovery
of how and why parallel processing

makes
the modern supercomputer **fastest**
changed the way
we think about how to build
the **fastest** computer.
It made the news headlines,
in 1989,
when I discovered
that we could execute
extreme-scale computational physics codes
and execute them **across**
an ensemble of 65,536
tightly-coupled commodity-off-the-shelf
processors
that shared nothing with each other
that were **identical**
and that were equal distances
apart
and that I visualized
as a new internet
that encircled a globe, or a hypersphere,
in sixteen-dimensional hyperspace.
My contributions

to extreme-scale computational mathematics
are mathematical inventions
in both algebra and calculus
that are mathematically speaking,
both **abstract** and **obscure**.
However, you benefit from
the new algebra and the new calculus
that I contributed
to mathematical knowledge.
The petroleum industry
use my contributions to extreme-scale
mathematical computations
and use the new mathematical knowledge
within the supercomputers
it uses to discover and recover
otherwise elusive
crude oil and natural gas.
My invention is the reason
one in ten supercomputers
are purchased by the petroleum industry.
My contributions
to computational mathematics
was the reason

I was invited to deliver a lecture 549
to research mathematicians
attending the International Congress
of Industrial and Applied Mathematics,
called ICIAM '91 in Washington, DC.
That congress
was the largest gathering
of mathematicians, ever.
To quote myself
from that ICIAM '91 lecture
that I gave on the morning of July 8, 1991
to the International Congress
of Industrial and Applied Mathematics,
I said:

[quote]
"In extreme-scale
petroleum reservoir simulations,
the governing system of
partial differential equations
of the modern calculus
are often anisotropic,
or have coefficients

with strong directional dependencies.
That is, my differential operators
are position dependent, non-axis aligned,
and have privileged spatial directions.
I invented
new partial **difference** equations
that I defined **across**
two and three-dimensional grids.
I used my new finite **difference** algorithms
to discretize
my new system of nine
coupled, non-linear, time-dependent,
and state-of-the-art
partial **differential** equations
of the modern calculus
that I also invented
in the early 1980s."
[unquote]

I continued my abstract lecture
of July 8, 1991

at the International Congress
of Industrial and Applied Mathematics
in Washington, DC:

[quote]
“The first tri-diagonal algorithm
that I investigated on a sequential processing supercomputer
back in 1974 in Corvallis, Oregon,
that was the first supercomputer
to be rated at
one million instructions per second
was a three-step adaptation of
Gaussian Elimination.
That algorithm converged to my solution
when my tri-diagonal system
is diagonally dominant.
That is, the norm of the diagonal entries
must be greater than
the sum of the norms
of the sub-diagonal
and super-diagonal entries.

[unquote]

552

17.1.5 Diary of an Extreme-Scale Arithmetician

I continued my abstract lecture
of July 8, 1991
at the International Congress
of Industrial and Applied Mathematics
in Washington, DC:

[quote]
“In the first step
for solving my tri-diagonal system
of equations of algebra,
I factorized the coefficient matrix
into lower and upper triangular matrices,
that were each bi-diagonal.
If the size of my linear system of equations
of algebra
was N by N,
or a million billion trillion
by a million billion trillion,
I performed, *a priori*, 2 times

a million billion trillion multiplications
plus a million billion trillion additions.
In the second step
for solving my tri-diagonal system
of equations of algebra,
I performed a forward substitution
on the lower triangular
bi-diagonal matrix.
My *a priori* computational workload
for this second step
was a million billion trillion multiplications
plus a million billion trillion additions.
In the third and final step
for solving my tri-diagonal system
of equations of algebra,
I performed a backward substitution
on the upper triangular, bi-diagonal matrix.
My *a priori* computational workload
was 2 times a million billion trillion
multiplications
plus N additions.
And my total *a priori*
computational workload

—in floating-point
arithmetical operation counts—
for all three steps
was five times a million billion trillion
multiplications
plus three times a million billion trillion
additions."
[unquote]

I mathematically discovered that
it's both logically and physically impossible
to massively parallel process
the solution of a **tri-diagonal** system
of equations of algebra
but it is possible to parallel process
the solution of a **diagonal** system
of algebra.
As an extreme-scale
computational mathematician
that was supercomputing
petroleum reservoir simulation codes
and supercomputing them **across**
my ensemble of tightly-coupled

commodity processors,
my end was not to merely solve
a **tri-diagonal** system of equations
of algebra.
We solve systems of equations of algebra
to discover and recover
otherwise elusive
crude oil and natural gas
that are buried one mile deep
inside oilfields,
such as the Niger Delta oilfields
of southeastern Nigeria.
In the 1970s and ‘80s,
I stepped backward
to the first principles, or the laws of physics.
I stepped backwards
to enable me to reformulate
nine partial differential equations
of modern calculus.
I discretized those
partial differential equations
to obtain an extremely-large
system of equations

of algebra
that's **diagonal**.
At the International Congress of Industrial
and Applied Mathematics,
and as a research computational
mathematician in quest for
never-before-seen mathematical equations,
I explained that I invented nine
never-before-seen
partial differential equations
of modern calculus.
That is, I invented
partial differential equations
that fit the second law of motion
of physics,
rather than invent
a law of motion of physics
that fit the partial differential equations
on the mathematician's blackboard
and in the calculus textbook.
Back on June 20, 1974,
at the Computer Center,
in Corvallis, Oregon,

this **Gaussian Elimination** algorithm
for solving a system of tri-diagonal
equations of algebra
was efficient
when the supercomputer scientist
must do only one thing at a time.
The **tri-diagonal** algorithm is efficient
for sequential processing supercomputers
because, in part, I did not have to store
the **off-diagonal zeroes**.
The **tri-diagonal** algorithm
was the workhorse
of algebraic computations
on a sequential processing supercomputer,
and remains so.
Due to its success,
computational mathematicians misconstrued
the **tri-diagonal** algorithm
as the end,
instead of as one of the means
to the end.
The end for an extreme-scale
computational physicist

that is parallel processing **across**
millions upon millions of
tightly-coupled commodity processors
is to attain a surer and deeper understanding
of how the universe works.
The end is to harness that new knowledge
of the partial differential equations
of planet Earth
to make planet Earth
a better place for human beings
and for all beings.
The end for an extreme-scale
computational geophysicist
is to use her petroleum reservoir simulator
as a digital divining rod
for discovering and recovering
otherwise elusive
crude oil and natural gas,
such as recovering
some of the 70 percent
of the crude oil and natural gas discovered
in **Oloibiri Oilfield**
in **Bayelsa State**, of southeastern Nigeria,

that began production in 1958
but was abandoned twenty years later
in 1978.
Back in the supercomputers of 1974
with very limited storage,
I stored the non-zero entries
of my **tri-diagonal** systems
as three vectors.
The **tri-diagonal** algorithm
is inherently sequential
and for that reason is a curse
to the programmer
of the modern supercomputer.
The **tri-diagonal** algorithm
is a curse because
the massively parallel processing
supercomputer scientist
wants to massively parallel process **across**
sixty-five thousand
five hundred and thirty-six [65,536]
tightly-coupled processors
that were identical
and that shared nothing with each other,

or to massively parallel process
across as many computers,
or to solve sixty-five thousand
five hundred and thirty-six [65,536]
algebraic problems **at once**.
The **tri-diagonal** algorithm
of the 1950s sequential processing
supercomputers
that was widely hailed as a blessing
for **alternating direction implicit** methods
used to **discover** and **recover**
otherwise elusive
crude oil and natural gas
became a curse in the early 1980s
when I tried to parallelize
the mathematically impossible to parallelize,
namely,
to massively parallel process
an inherently sequential algorithm
by solving many algebraic problems
at once, or in parallel.

17.1.6 From Mathematics to Magic

Abstract mathematics
is an art on the blackboard
and an experimental science
on the motherboard.
I am an experimental mathematician
of the 1970s and '80s
who used his system of
coupled, non-linear, time-dependent,
and state-of-the-art
partial differential equations
of modern calculus,
called Philip Emeagwali's equations.
I used that system of equations
of calculus and algebra
to invent
the massively parallel processing
supercomputer.
Extreme-scale computational mathematics
becomes magic
when it is executed **across**
two-raised-to-power sixteen tightly-coupled

commodity-off-the-shelf processors
that shared nothing with each other
that are identical
and that are equal distances
apart.
I invented
how to solve the toughest problems
arising in modern algebra
and how to solve those problems **across**
my ensemble of 65,536 processors
and how to solve them
when my ensemble emulates
one seamless, cohesive
massively parallel processing supercomputer
that was the precursor
to the modern supercomputer.
That new supercomputer
is not a computer *per se*.
That new supercomputer
is a new internet *de facto*.
I experimentally discovered
the precursor to the modern supercomputer
and I discovered it

at 8:15 on the morning
at Los Alamos
(New Mexico, United States) time
of Tuesday the Fourth of July 1989.
That discovery
put me in the news headlines
of major U.S. newspapers
as the African supercomputer wizard
that won top US prize
and won it for his contributions
to the development
of the massively parallel processing
supercomputer.
Behind each of those 65,536
tightly-coupled processors
that outlined a new internet
was a lone wolf research
supercomputer scientist
that was not a member
of a 400-person research team
that coded and verified
each line of his massively parallel processing
instructions and **invented**

each partial **difference** equation
of his extreme-scale algebra.
That Grand Challenge Problem
of mathematics and supercomputing
had to be solved **across**
my ensemble of 65,536 processors.
Each processor within my ensemble
operated its own operating system.
Each processor
has its own dedicated memory
that shares nothing with each other.
I am the research supercomputer scientist
that **invented**
each of those partial **differential** equation
of modern calculus
from which those
partial **difference** equations
of algebra arose.
I—**Philip Emeagwali**—was that research
massively parallel processing
supercomputer scientist
that ensured that
his two-raised-to-power sixteen,

or 65,536, message-passed
supercomputer codes
were one-to-one mapped
to 65,536 sixteen-bit-identified processors
and ensured that the codes and processors
came together
as the fastest and never-before-seen
supercomputer
that made the news headlines
across major U.S. newspapers
and that was reported
in the June 20, 1990 issue
of the *Wall Street Journal*.

17.2 Contributions of Philip Emeagwali to Computational Mathematics

17.2.1 On Large-Scale Algebra

I am the research
computational mathematician

that invented
how to harness
the potential aggregate
supercomputing power
of an ensemble of
64 binary thousand processors.
I invented
how to use that ensemble—that is a
new supercomputer—to solve
the toughest problems in linear algebra
that arose from
partial **difference** approximations
of initial-boundary value problems
of modern calculus.
I invented
how to solve extreme-scaled
systems of equations
of linear algebra
that arises when simulating
the subterranean motions
of crude oil, injected water, and natural gas
that flow a mile-deep
inside a petroleum reservoir,

such as inside the Niger-Delta oilfields
of the southeastern region of Nigeria.
Research computational mathematicians
of the early 1950s,
responded to the invention
of the sequential processing supercomputer
of 1946
and they did so by inventing
algebraic algorithms
that were efficient
when they were sequentially processed
within only one electronic brain
of their supercomputer.
Research computational algebraist
of the early 1950s responded
to the grand challenge
of simulating petroleum reservoirs
and they did so by inventing
a system of equations
of linear algebra
that were **tridiagonal**.
They were called **tridiagonal** because
the associated matrix

of the **tridiagonal** system
is banded near the main diagonal
and only has nonzero elements
on the main diagonal
and nonzero elements
on the first diagonal
that was below the main diagonal
and nonzero elements
on the first diagonal
that was above the main diagonal.
In reality, the system of equations
of linear algebra
that originated from a system of
partial differential equations
of modern calculus
could be formulated
to be fourth order accurate.
Such fourth order accurate schemes
arises from five-stenciled
finite difference discretization schemes
that yield a system of equations
of linear algebra
that are **penta**diagonal,

namely, have five nonzero bands
around and along the main diagonal.
Such five-point finite difference schemes
are rarely used because
they are more complex
and demands greater
floating-point arithmetical operations.
For these reasons,
computational mathematicians
prefer the second order of accuracy
that arises from three-stenciled
finite difference approximations.
The numerical solutions
of extreme-scaled problems
in linear algebra
is the recurring decimal
in all supercomputers, past and present.
The August 25, 1947 issue
of *The New York Times*
carried an article that was titled:
"New Giant 'Brain' Does Wizard Work."
That *New York Times* article explained that:

[quote]
“...the machines under construction
will have a ‘built-in intelligence’
which will enable them to handle
the most complicated differential equations
of physics and engineering,
performing hundreds of
separate mathematical operations
without the intervention
of a human operator...”
[unquote]

My invention
that occurred on the Fourth of July 1989
was an invention
of a new supercomputer
with ‘built-in intelligence,’
namely, sending and receiving
email messages
and doing so **across**
my ensemble of 65,536 processors.
That new supercomputer
enables the computational mathematician

to solve the toughest problems
arising in modern calculus,
such as solving the system of
coupled, non-linear, time-dependent,
and state-of-the-art
partial differential equations
that arises in extreme-scale
computational fluid dynamics,
such as the general circulation models
that has rigorous reproducibility
requirements that are used to **foresee**
otherwise unforeseeable climate changes
and such as the
petroleum reservoir simulators
that must be used to discover and recover
otherwise elusive
crude oil and natural gas,
including those from
the Niger-Delta oilfields
of southeastern region
of my country of birth, Nigeria.
The one mile deep Niger-Delta oilfield
doesn't fit into a laboratory,

or into one computer.
For that reason, I invented
how to fit
the Niger-Delta oilfield
into a new supercomputer
that is not a computer *per se*
but that is a new internet *de facto*.
My new internet
is powered by my ensemble of
65,536 commonly-available processors.
Each processor within my ensemble
operated its own operating system.
Each processor
had its own dedicated memory
that shared nothing with each other.
My invention
of the massively parallel processing
supercomputer
that occurred
on the Fourth of July 1989
was reported in the June 20, 1990 issue
of *The Wall Street Journal*.
Eleven years later,

that invention
of a new supercomputer
was reconfirmed
by then President Bill Clinton
and reconfirmed
in his presidential lecture
of August 26, 2000
that was delivered
to the Nigerian parliament
in Abuja, Nigeria.

17.2.2 Mathematics is Living Knowledge

The petroleum industry mathematician
marched forward
with her sequential algorithms.
Her sequential algorithms
accurately solved
the **tri-diagonal** system of equations
that approximated
the system of governing
partial differential equations

that, in turn, did not accurately represent
the **actual** initial-boundary value problem
that was at the mathematical core
of the petroleum reservoir simulator
that was used to discover and recover
otherwise **elusive**

crude oil and natural gas.
That is, accurately solving
a tri-diagonal system of equations
that erroneously approximated
a production crude oil and natural gas field
in the Niger Delta region
of Bayelsa State
of the southeastern region of Nigeria
amounts to metaphorically asking
each member of my ensemble of
65,536 processors
the wrong questions
and expecting the right answers
from each processor
that was asked the wrong questions.
For that reason,

I began developing
my computational physics codes
from first principles, namely,
the Second Law of Motion
of physics
and I correctly re-derived
the system of partial differential equations
that governs
any production crude oil
and natural gas field.
I did not want to mathematically
march forward
and do so with the classic, much beloved
but **erroneous**
alternating direction implicit methods
that were invented
by research mathematicians
of the early 1950s.
In my mathematical nightmares,
I envisioned extreme-scale
numerical algebraists
mathematically marching forward
and doing so along my one million

forty-eight thousand
five hundred and seventy-six [1,048,576]
bi-directional email wires
and mathematically marching
to my sixty-five thousand
five hundred and thirty-six [65,536]
commodity-off-the-shelf processors
that defined my new internet
and redefined it
as a new supercomputer.
The reason I rejected
the much beloved tri-diagonal system
of equations of modern algebra
was that it was an algebraic bye-product
of the *de facto* erroneously reiterated
formula,
Force is **not [**is **not,** is **not]**
equal to **Mass** times **Acceleration.**
Put differently,
the alternating direction implicit method
was a succession of leaps (or steps)
that succeeded from
one error in the x-direction

to a similar error in the y-direction
and to a similar error in the z-direction.
The alternating direction implicit method
of extreme-scale algebra
and petroleum reservoir simulation
spewed erroneous computational results
and did so along sixteen directions
and spewed those errors
into my two-raised-to-power sixteen,
or 65,536, tightly-coupled processors
that shared nothing with each other.
In 1981, I was the research
massively parallel processing mathematician
in College Park, Maryland, United States,
that mathematically discovered
that for the three decades
onward of the early 1950s
that research mathematicians
developed abstract
partial differential equations
and developed them
for the crude oil and natural gas industry
and the mathematician's abstract equations

became deeper abstraction
for the sake of abstraction.
I mathematically discovered
that the research mathematician’s fancy
system of partial differential equations
of modern calculus
encoded only three forces,
instead of encoding the actual four forces
that drives crude oil and natural gas
towards the production crude oil
and natural gas wells.
I mathematically discovered
that none of those **abstract**
partial differential equations
of modern calculus and petroleum physics
equated
with the subterranean motions
of the crude oil and natural gas
that were flowing a mile-deep
inside the production oilfields
they were supposed to model.
I was outraged that for
nearly a century and half,

onwards of the discovery
of Darcy's formula in 1856,
that research mathematicians
who developed
what should be the most important equation
in modern mathematics
and computational physics
were *de facto*
contemplating their navels
and forgetting that
mathematical physics
should be a living body of knowledge
that ensures that
the mathematician and the physicist
should be congruent to each other,
or be like twins of the opposite sex.
Mathematical **abstraction**
does not equate to **functionality**
and **accuracy**.
In 1981 and while in College Park,
Maryland, United States,
I was the research
massively parallel processing

computational mathematician
that demanded that the forces
encoded into the abstract
partial differential equations
of the petroleum industry
should be congruent
with the inertial, pressure, viscous,
and gravitational forces
that drove the crude oil, injected water,
and natural gas
and drove those fluids
from the water pumping well
to the crude oil
and natural gas production well.
Metaphorically speaking,
the partial differential equations
of textbooks on petroleum physics
were built on **sand**,
not built on oil **sand**.
And equations built on sand
produces verbiage
and are used to compute cycles of errors.
Across my ensemble of 64 binary thousand

tightly-coupled processors
it can be said that
Garbage In, Garbage Out.
The alternating direction implicit method
became useless
because it propagates critical errors
which, in turn, reduces the amount of
otherwise **elusive**
crude oil and natural gas
that could be discoverable and recoverable.

17.2.3 Algebra Across Philip Emeagwali Internet

The core essences
of my computational experiments
were to email questions and answers
that pertained to those equations
and algorithms,
that pertained to those
partial **differential** equations
of modern calculus
and computational physics
and partial **difference** equations

of modern algebra
and that were generated
within each of my 65,536 tightly-coupled
commodity-off-the-shelf processors
that had sixteen orthogonal pathways
and that were identical
and that were equal distances **apart**
and to email each processor
via email wires
that metaphorically
had a one-to-one correspondence
to the 1,048,576
bi-directional edges of the cube
in a sixteen-dimensional universe
that I visualized as **etched**
onto the surface of a sphere
in a sixteen dimensional universe
and that I visualized
as a new global network processors
and email wires
that had no center, no edge.
Those emails delivered my
65,536 computational physics codes

and delivered them
to 65,536 tightly-coupled processors
that shared nothing with each other
that had a **one-to-one** correspondence
to the two-raised-to-power sixteen,
or sixty-four binary thousand,
or 65,536, vertices
of the same hypercube in hyperspace.
To me, **Philip Emeagwali**,
my theory of parallel processing
was a metaphor
for the lyrics or screen play,
while my experiments
across 65,536 processors
represented the song or play.
I am the mathematical physicist
that experimentally discovered
on the Fourth of July 1989
how to harness
sixty-four binary thousand processors
and invented
how to use those processors
to process codes

arising in extreme-scale
computational physics
and invented
how to communicate
those physics codes
and invented
how to do so at email speeds
that were previously unrecorded.
As that research
computational mathematician
that was in major U.S. newspapers
onward of 1989,
I had the visceral feeling
that I wrote the lyrics of the song
for each of the sixty-four binary thousand processors
that outlined and defined
the **Philip Emeagwali** internet
that is a new supercomputer *de facto*.
I wrote the lyrics
and then I sang it
on the Fourth of July 1989
that was the U.S. Independence Day.

I had the visceral feeling
that I wrote the screen play
of a computational physics movie
with sixty-four binary thousand physicists,
each a metaphorical dancer,
that metaphorically danced **across**
one binary million pathways.
I had the visceral feeling
that I was the dance choreographer
that acted in his production,
which in my reality
was a movie
that is a petroleum reservoir simulation
of computational physics.
I visualized my 65,536
computational physics codes
as metaphors for as many screen plays.
If printed on paper,
my screen play would weigh
eighty million pages
of arithmetical data!
That was the amount of information
that the equations and algorithms

that I invented
generated from each supercomputing cycle
of computations
at sixty-four binary thousand processors.
That was the amount of information
that I communicated
along email wires
that had a one-to-one correspondence
to the one binary million
bi-directional edges of the hypercube
in the sixteenth-dimensional hyperspace.
For my discovery
of the Fourth of July 1989
that was recorded by the news media
and recorded in the June 20, 1990 issue
of the *Wall Street Journal*,
each supercomputing cycle lasted
one seventh of a second.
I invented
how to send emails—and how to send them
across
that new internet
that **Philip Emeagwali** internet

and how to send them with
five-subject lines.
I invented
how to send an email
that will be received
at a unique sixteen-bit address
that had no @ sign
or dot com suffix.
I invented
how to swap and distribute
my questions and answers
into and from two-to-power sixteen
commodity-off-the-shelf processors.
I invented
how to send them along
two-to-power twenty
email wires.
It's not enough that I know the
Philip Emeagwali internet.
That new internet
must know **Philip Emeagwali**
as its sole programmer.
To squeeze the maximum

computational speed
out of my ensemble of
65,536 processors
demanded that I understand my ensemble
as a new global network of processors
that is a new internet
that is the **Philip Emeagwali** internet.

17.2.4 Quest for Crude oil and natural gas

In **Oloibiri oil field**
of Bayelsa State
of the southeastern region of Nigeria,
the crude oil that was discovered
back in 1956
was trapped one mile
underneath the surface of the Earth.
Crude oil is a fossil fuel
that formed from the remains of
algae and zoo**plankt**on
and from tiny prehistoric plants and animals
that died hundreds of millions of years ago
and that were buried

at the bottom of the sea.
The **Oloibiri oil field**
that became operational in 1958
was formed hundreds of millions
of years ago.
The **Oloibiri oil field**
was abandoned in 1978
and abandoned twenty years
after it became operational.
Only about twenty percent
of the crude oil discovered in **Oloibiri**
was recovered.
One in ten
parallel processing supercomputers
were purchased by the petroleum industry
and were used to solve
extreme-scale problems
arising in **algebra**
that must be solved as a pre-condition
to discovering and recovering otherwise
elusive
crude oil and natural gas.
Accurately predicting the performance of

crude oil and natural gas reservoirs
in the Niger Delta region of Nigeria
translates to more drilling decisions.
With today's parallel processing
supercomputer technology,
more crude oil and natural gas
would have been recovered
from **Oloibiri oil field**,
if and only if we had solved
the system of trillions upon trillions
of equations of algebra
that would have arisen
if the parallel processing supercomputer
of today
was available back in 1958
and was used to predict
the subterranean motions
of crude oil, injected water, and natural gas
that flowed a mile deep
inside the production oilfields
of the Niger Delta region
of the southeastern region of Nigeria.
We cannot see, hear, or feel

591

the subterranean motions of the crude oil
and natural gas
that are flowing one mile deep
underneath our feet.
The supercomputer simulation
of the subterranean motions
of the crude oil and natural gas
that are flowing one mile deep
enables the petroleum geologist
to see—with his digital eyes—
the flow patterns
of the crude oil and natural gas
that are invisible to our naked eyes.
The massively parallel processing
supercomputer
that solves the trillions upon trillions
of equations of **algebra**
that arises
from the **extreme-scale**
petroleum reservoir simulator
is the new age divining rod
that must be used
to discover and recover

otherwise elusive
crude oil and natural gas.

17.2.5 Discovering the Elusive

Back in 1989,
it made the news headlines
that an African Supercomputer Wizard
had won the US top prize
for inventing
the massively parallel processing
supercomputer
and for inventing
how to breakup **extreme-scale**
problems arising in algebra
and for inventing
how to break up their computation-intensive
floating-point arithmetical problems
into 65,536
less computation-intensive
arithmetical problems.
I—**Philip Emeagwali**—is that African
massively parallel processing

supercomputer scientist
who invented
how to solve those smaller problems
and invented
how to solve those problems in parallel
and invented
how to solve those problems **across**
a new internet
that I visualized
as a new global network of
65,536 tightly-coupled
commodity processors
that shared nothing with each other.
My invention
of how to parallel process
and how to do so **across** a new internet
is used to accelerate scientific discovery
in the fields of
computational fluid dynamics,
such as executing
excruciatingly-detailed
petroleum reservoir simulation
and executing them

to discover and recover otherwise elusive
crude oil and natural gas
and such as executing
excruciatingly-detailed
general circulation models
and executing them
to foresee otherwise **unforeseeable**
climate changes.
The necessity
to execute these extreme-scaled problems
arising in computational physics
is one of the technological grand challenges
that **stimulated**
the development of the parallel processing
supercomputer.

17.2.6 Changing the Way We Look at the Computer

The experimental discovery
of parallel processing
changed the way we look
at the computer,
changed the way we look

at the supercomputer,
and changed the way we solve
the toughest problems arising in
extreme-scale algebraic computations.
Parallel processing
is the paradigm shift of tectonic proportions
in the history of computing
that changed the way
crude oil and natural gas
are discovered and recovered.
Parallel processing
changed the way
high-resolution, long-running
general circulation models
for otherwise unforeseeable
global warming are foreseen.
Parallel processing
changed the way
computational physics is done.
The toughest problems
in extreme-scale computational physics
are linked by a common thread, namely,
the modern supercomputer

that parallel processes
their extremely computation-intensive
floating-point arithmetical computations
and execute them **across**
an ensemble of up to ten million
commodity-off-the-shelf processors.
As a research supercomputer scientist
of the 1970s and '80s,
my quest was to invent
how to harness
my ensemble of 65,536
commodity-off-the-shelf processors
and to also invent
how to use those processors
to bring the toughest problems
in extreme-scale algebra
to their knees
and to invent
how to enable the petroleum industry
to use the massively
parallel processing supercomputer
to chase otherwise unrecoverable
crude oil and natural gas.

My discovery
of massively parallel processing
that occurred at
10:15 on the morning Eastern Standard Time
Tuesday the Fourth of July 1989
that was the U.S. Independence Day
made the news headlines
in 1989, and thereafter,
and emboldened the call to action
for computational geophysicists
to embrace
the technology of the
massively parallel processing
supercomputer.
My discovery represents a new way
of looking at the computer.
For my country of birth, Nigeria,
the fastest supercomputer
is a poverty alleviation technology.
Crude oil and natural gas are recovered
after using the massively parallel processing
supercomputer technology
to simulate countless crude oil

and natural gas
recovery scenarios.
In 1989,
I was in the news
for experimentally discovering
how to harness
the massively parallel processing
supercomputer
and for discovering
how to use the new technology
to reduce the **time-to-solution**
for solving
extreme-scaled system of equations
of **algebra**
and how to reduce that **time-to-solution**
from 180 years, or 65,536 days,
to only one day of **time-to-solution**.
I was in the news because
reducing that **time-to-solution**
increases the odds of
discovering and recovering
otherwise elusive
crude oil and natural gas.

That discovery
of the massively parallel processing
supercomputer
that occurred
on the Fourth of July 1989
was praised as a giant leap forward
in the development of the modern algebra.
That discovery
of parallel processing
is my contribution to algebra.