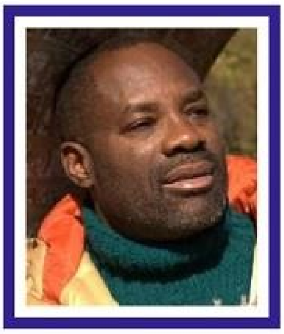
16 An African Mathematician and His Contributions to Mathematics



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

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

The need to calculate
is as old as humanity.

The need to compute
existed because it is central to
human existence.
The Latin equivalence
of the word “computer”
was first used in print
two thousand years ago.
The word “computer”
was first used by the Roman author
Pliny the Elder.
The word “supercomputer”
was coined in 1967.
I believe that our children’s children
will coin a new word
for their supercomputers.
I believe that our children’s children
will invent supercomputers
that are science fiction to us.

16.1 Contributions of Philip Emeagwali to Algebra

The discovery and recovery
of every single barrel of oil
from any oilfield
in the Niger-Delta region of Nigeria
must be preceded
by the massively parallel processed solution
of the toughest problem
in extreme-scale algebra.
For the fifteen years, inclusive
of from the 20th of June of 1974
that I was in Corvallis, Oregon, United States
through the Fourth of July 1989
in Los Alamos, New Mexico, United States,
I conducted my supercomputer research,
and did so from speeds of
one million instructions per second
in Oregon
to **billions** upon **billions**
of floating-point arithmetical calculations

that I executed **across**
a new internet
that is outlined and defined
by a new global network of
two-raised-to-power sixteen
tightly-coupled processors
that are commonly available
in the market.
Each of those 64 binary thousand
commodity processors
operated its own operating system.
Each of those 65,536 processors
has its own dedicated memory
that shares nothing with each other.
I was in the news in 1989
because I invented
how to solve the most
computation-intensive problems
arising in large-scale
algebraic computations.
Such problems arose from discretizing
partial differential equations
that, in turn, arose from

physics-based supercomputer simulations
of the motions of fluids
that flow below the surface of the Earth,
such as the mile-deep
crude oil, injected water, and natural gas
in the Niger-Delta region of Nigeria;
and from supercomputer simulations
of the motions of fluids
that flow on the surface of the Earth,
such as the River Niger, Lake Chad,
and the Atlantic ocean;
and from supercomputer simulations
of the motions of fluids
that flow above the surface of the Earth,
such as atmospheric rivers
that are defined
as bands of moisture in the sky
that can discharge as much water
as many land rivers.
Following my invention
of the massively parallel processing
supercomputer
that occurred

on the Fourth of July 1989,
and that occurred in Los Alamos, New
Mexico, United States,
the top publications
in petroleum engineering and mathematics
credited I—**Philip Emeagwali**—
with the invention
of how to solve the toughest problems
arising in extreme-scale algebra
that must be solved
as a pre-condition for discovering
and recovering
otherwise elusive
crude oil and natural gas.
For fifteen years,
my supercomputer research on how to solve
the toughest problems
arising in large-scale algebraic computations
and how to solve them **across**
a new internet
that is a new ensemble of 65,536
commonly-available processors
that were identical

and that were equal distances **apart**
was criticized, scorned, and rejected.
My invention
of how to solve the most extreme-scale
problems arising in algebra
and how to solve them **across**
my new internet
that is a new global network of
65,536 commodity processors
was only accepted
after the Fourth of July 1989,
the date that I invented it.
To the non-mathematician,
my mathematical invention is abstract
and impenetrable.
But my contributions
to calculus and algebra
made sense to the research mathematician.
For that reason, my contributions
to using the modern supercomputer
to solve the toughest problems
arising in calculus and algebra
was the cover stories

of top mathematics publications,
such as the May 1990 issue
of the *SIAM News*.
The *SIAM News*
is the flagship publication
of the mathematics community.
The *SIAM News*
is published by the Society for
Industrial and Applied Mathematics.
I was not on the cover
of top mathematics publications
because I was good looking.
I was on the cover
of top mathematics publications
because I contributed
new algebraic knowledge
to the existing body of knowledge
written in algebra textbooks,
namely, I invented
how to solve the toughest problems
arising in algebra and calculus
and invented how to solve them **across**
my new internet

that is a new global network of processors
that emulates one seamless, cohesive
new supercomputer.

16.1.1 Who Invented the Quadratic Equation?

My first introduction to algebra
was in mid-1964
in the fifth grade
in Agbor, in the Midwestern Region
of Nigeria.
My first algebra textbook
was written by an English schoolmaster
named Clement Vavasor “CV” Durell.
Who invented the quadratic equation
of algebra?
The quadratic equation
is one of the most important contributions
to mathematics
that was made by Africans
and made in Africa.
The quadratic equation
was first solved

in the **Berlin** Papyrus.
The **Berlin** Papyrus
was excavated in Africa
but renamed after **Berlin**, Germany.
In the history of mathematics,
it was the tradition
credit the mathematical knowledge
that was developed in Africa
to Europe.
Following that tradition
of historians of mathematics,
the **Berlin** Papyrus
was credited to the city of **Berlin**, Germany
and not named after
the city in Africa
that the Papyrus was excavated from.
The [quote unquote] “**Berlin** Papyrus”
was written in Africa
and written by Africans
and written between
three thousand seven hundred and twenty
[3,720] years ago
and four thousand one hundred and eighty

[4,180] years ago.
In that African Papyrus
that was written four thousand years ago,
the quadratic equation
was stated as
x squared plus y squared
equals 100
and y equals three-quarters
divided by the product of four and x.

16.1.2 Contributions of Philip Emeagwali to Algebra

The quadratic equation of algebra
that is in the homework assignment
of a 12-year-old
contains at least one unknown variable
that is squared.
That unknown variable
is often called x.
The 24 million system of

partial **difference** equations
of algebra
that I solved
on the Fourth of July 1989
and solved **across** a new internet
that is a new global network of
65,536 tightly-coupled,
commonly available processors
that made the news headlines
contained 24 million unknown variables.
That was a world record in algebra.
In my Fourth of July 1989
invention,
I computed **across** a new internet.
I visualized my new internet
as my 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
that shared nothing with each other.
Each processor within my new internet

had one hundred and thirty-one thousand
and seventy-two [131,072] bytes of memory
that I used
for the **integer** and **logical** operations
that arose
from my algebraic computations.
I also used
two thousand and forty-eight [2,048]
floating point **co-processors**.
I used those co-processors
to execute
my floating point arithmetical computations.
Each co-processor
had four megabytes of memory.
The frontier of extreme-scale algebra
is far beyond the quadratic equation
that contains two unknown variables
that was first solved
four thousand years ago
in Africa.
Extreme-scale algebra
arise during the general circulation modeling
that must be used to foresee

otherwise unforeseeable
global warmings.
Extreme-scale algebra
is used to discover and recover
otherwise elusive
crude oil and natural gas.
Extreme-scale algebra
is used to solve the toughest problems
arising in computational physics.
The most important quest
in algebra
was to theoretically
and experimentally invent
how to solve extreme-scale
tri-diagonal system of equations
of algebra
and how to solve those equations **across**
millions upon millions
of 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.
One reason I was the cover stories
of mathematics publications
was because
I mathematically discovered
that it’s **impossible**
to reduce a tri-diagonal system of equations
of algebra
and reduce it
to a mathematically equivalent
diagonal system of equations
of algebra.
I, *de facto*, circumvented that **impossibility**
by taking two steps backward
and to the granite core
of the physical problem
that is mathematically modeled,
namely, the extreme-scale
petroleum reservoir simulation,
that was governed by
a tri-diagonal system of equations
of algebra
that arose from discretizing

a system of partial **differential** equations
of modern calculus
and arose from approximating
that system
by a companion system
of partial **difference** equations
of a new algebra.
It was at that granite core
of the most excruciatingly-detailed
petroleum reservoir simulation
that I reformulated
the partial **differential** equations
that is a new calculus
that was not in any calculus textbook
that was printed in the 1970s or earlier.
The reason my contributions to mathematics
was the cover story
of the top mathematics publication, namely,
the May 1990 issue
of the *SIAM News*
of the Society of Industrial
and Applied Mathematics
was that I created a new calculus

and I did so
by inventing thirty-six [36] additional
partial derivative terms
of calculus.
In the old calculus
that is used to recover
otherwise **unrecoverable**
crude oil and natural gas,
only forty-five [45]
partial derivative terms
represented the Second Law of Motion
of physics
that was **encoded**
into the three-phased—crude oil,
injected water, and natural gas—
extreme-scaled
petroleum reservoir simulator.
My contributions
to the body of mathematical knowledge
called calculus and algebra are these:
I added the thirty-six [36]
partial derivative terms
of modern calculus

that I invented
to the forty-five [45]
partial derivative terms
that prior research mathematicians
invented.
And I did so to bring the total
partial derivative terms
to eighty-one (81).
In the advanced calculus textbooks
and in the old paradigm
of extreme-scale algebraic computations
arising from
petroleum reservoir simulations,
the governing system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern mathematics
was parabolic, instead of hyperbolic.
In my new paradigm
of extreme-scale algebraic computations,
that enabled me to
massively parallel process
and to process **across** a new internet

that I visualized
as a new global network of
65,536 tightly-coupled,
commonly available processors
that shared nothing with each other
and in that new paradigm
my system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern calculus is **hyperbolic**,
not **parabolic** as written in
calculus textbooks.
My contributions to calculus
was a paradigm shift because
such governing
partial differential equations
are described in all the textbooks
as **parabolic**, instead of **hyperbolic**.
They are described in calculus textbooks
on the **multi-phase** and **multi-physics**
porous media flow
of crude oil, injected water, and natural gas,
such as the crude oil and natural gas

flowing from the water injection wells
to the crude oil and natural gas wells
across the oilfields
of the Niger Delta region
of southeastern Nigeria.

16.1.3 Predicting Algebraic Error Propagation

Back in the early 1980s,
I was in College Park, Maryland.
I was a research computational
mathematician in College Park.
The contributions to mathematics
that I made in the early 1980s
were the cover stories
of the top mathematics publications,
such as the May 1990 issue
of the *SIAM News*
that was published by
the Society of Industrial
and Applied Mathematics.
My quest
for new mathematical knowledge

was in the fields of calculus and
numerical analysis.
My greatest focus
was in the field of extreme-scaled
computational mathematics.
I focused on never-before-seen ways
of using the most abstract
and the most advanced
arithmetical knowledge
and using that knowledge
to solve the toughest problems
that arise in computational physics
that otherwise will be unsolveable.
My new field
—of never-before-seen
extreme-scaled computations—
was at the **crossroad**
between calculus and algebra
and between arithmetic
and the computer.
I redefined my numerical analysis
as between
the singular processor

that computes sequentially
and my ensemble of processors
that computes **in parallel**,
or that solves many problems
at once.
In 1989, it made the news headlines
that an African supercomputer wizard
in the United States
had invented
how to use 65,536 processors
to solve as many problems in parallel.
I am that African supercomputer scientist
that was in the news
onward of 1989.
I was in the news because I invented
how to use one billion processors
to solve one billion
initial-boundary value problems
of calculus
and how to solve them in parallel
and, specifically, how to compute **across**
a new internet
that is a new global network of

two-raised-to-power sixteen
commodity-off-the-shelf processors
that were married together
as one seamless, cohesive
massively parallel processing supercomputer
and married together
by sixteen times as many
email wires.
The mathematical analysis
that was at the theoretical foundation
of my extreme-scale
computational mathematics
and that preceded
my invention
of the massively parallel processing
supercomputer
is called
the stability analysis
of **finite difference discretizations**
of partial differential equations
of modern calculus.
Stability analysis
was the extremely rigorous

and the analytical procedure
that I used to derive *a priori* error estimate
of the rate of propagation
of initial errors
and the rate as I computed forward in time
and computed within each processor
and communicated **across**
my new ensemble of 65,536
tightly-coupled, commodity processors
with each processor
operating its own operating system
and with each processor
having its own dedicated memory
that shared nothing with each other.
After going through some dense
and abstract stability analyses
in the early 1980s
and after conducting companion
computational experiments,
I mathematically discovered
that explicit finite difference
algebraic approximations
of the governing system of

partial differential equations
of modern calculus
that include the thirty-six (36)
new partial derivative terms
that I invented
allow longer computational time-steps
which, in turn, makes my calculations
faster.
That mathematical invention
was how I greatly reduced
the vexing time-step limit
that textbooks on computational physics
describe as the Courant Condition.
That Courant Condition
is the necessary condition
for the convergence
of the numerical solution
of an explicit
partial **difference** equation
to the analytical solution
of the original partial **differential** equation
that it was approximating.
That mathematical invention

was how I bypassed
the empirical Darcy’s formula
that was outdated and invented
back in 1856.
That mathematical invention
was how I replaced
a system of nine **algebraic** Darcy’s equations
that must be used by the petroleum industry
to describe
the subterranean motions
of multi-phased fluids.
I invented and replaced
those nine **algebraic** Darcy’s equations
with my more rigorous
system of nine partial differential equations
of a new calculus
that I invented
from first principles,
or from the Second Law of Motion
of physics.
Henry Darcy’s Law
is a statement in the fluid dynamics
of flows **across** a porous medium.

Henry Darcy's Law
states that the velocities of crude oil,
injected water,
and natural gas flowing **across**
the permeable Niger Delta oilfields
of southeastern Nigeria
is due to the difference in pressure.
Henry Darcy's Law
states that the velocities
of the crude oil, injected water,
and natural gas
are proportional to the pressure gradients
in the direction
of the flows of crude oil, injected water, and
natural gas.
That mathematical invention,
called Philip Emeagwali's Equations,
was how I bypassed
the vexing
eight processor limit
known as **Amdahl's Law**
that limits the number of processors
that should be incorporated

into massively parallel processing
supercomputers.

16.1.4 Contributions of Philip Emeagwali to Algebra

The first automatic
and sequential processing supercomputer
that was programmable
was invented in 1946.
That first supercomputer
was invented
to be programmed to solve
a large system of equations
of algebra
that arose from the
finite difference discretizations
of ordinary differential equations
of modern calculus
that, in turn, encoded
a set of laws of physics.
What made the sequential processing
supercomputer of 1946 unique
was that it computed automatically

and was, therefore, programmable.
Fast forward twenty-eight [28] years
from that first supercomputer,
and to June 20, 1974,
in Corvallis, Oregon,
and I was programming
the first supercomputer
that could execute
over one million instructions per second.
I used that first supercomputer
to solve the largest-scale problems
arising in modern algebra.
Fast forward fifteen [15] years,
and to the Fourth of July 1989,
I was in a dozen supercomputer centers
across the United States
and I was programming
the first massively parallel processing
supercomputer
that could execute
billions of calculations
and execute them **across**
my ensemble of up to

65,536 tightly-coupled processors.

16.1.5 Why is Philip Emeagwali Famous?

My invention
of a new supercomputer
put me in the news headlines
and in the June 20, 1990 issue
of the *Wall Street Journal*.
I was the cover story
of the June 1990 issue
of the *SIAM News*.
The *SIAM News*
is the top mathematics publication
and is published by
the Society of Industrial
and Applied Mathematics.
The cover stories in the *SIAM News*
report new inventions in mathematics
and they are written by
research mathematicians
and written for research mathematicians.
In the cover story of the *SIAM News*

of June 1990,
it was reported that I—**Philip Emeagwali**—
had mathematically invented
how to solve the toughest problems
arising in modern calculus
and arising in extreme-scale algebra
and invented how to solve them **across**
a new ensemble of 65,536
commonly available processors.
I invented
how to use that new supercomputer
to solve many problems **at once**
and to solve the largest-scaled problems
arising in modern algebra.

16.1.6 Contributions of Philip Emeagwali to Physics

On the Fourth of July 1989,
the state-of-the-art of that toughest problem
in modern algebra
was a system of 24 million equations
of algebra

that arose from
my finite difference discretizations
of a system of partial differential equations
that I invented
that mathematically encoded
a set of laws of physics
that governs
the subterranean motions of crude oil,
injected water, and natural gas
that flows one mile-deep
underneath the surface of the Earth
and that flows from water injection wells
towards
crude oil and natural gas production wells.
I visualized my new instrument
of computational physics
as a new internet
that I defined
as my new global network
of 65,536 tightly-coupled
commodity-off-the-shelf processors
with each processor
operating its own operating system

and with each processor
having its own dedicated memory
that shared nothing with each other.
I visualized my new internet
as a new instrument
for solving
the most extreme-scaled
problems arising in algebra
and for solving them
as one seamless, cohesive unit
that is a new supercomputer *de facto*.
The defining feature
of my invention
of that new internet
was that the new technology
enabled me to compute **synchronously**
and to communicate **automatically**
and to do so via emails
that I sent to and received from
two-to-power-sixteen
sixteen-bit long email addresses.
Each of my 64 binary thousand
email addresses

had no @ sign or dot com suffix.

16.1.7 Sometimes, the Impossible is Possible

Back in 1989,
no author of any mathematics textbook
understood the concept of
solving many problems **at once**,
or in parallel.
Back in 1989,
Seymour Cray
was the spokesman
for the supercomputer community.
If **Seymour Cray's** granddaughter
came to him for help
with her homework assignment
on how to solve
many mathematical problems **at once**,
Seymour Cray
would not have been able
to help her
to solve her problem in parallel.

The reason was that **Seymour Cray**
ardently believed that
the supercomputer technology
that I invented
that enables parallel processing **across**
my ensemble of 65,536
tightly-coupled processors
was impossible and science fiction.

16.2 A Day in the Life of a Black Mathematician

16.2.1 Quest of Philip Emeagwali for New Mathematical Physics

I was an apprentice supercomputer scientist
for forty-five [45] years.
After investing forty-five [45] years
in the craft of supercomputing,
I had the command of materials

that I needed to deliver
my supercomputer lectures
in prose and poetry,
rather than on blackboards and power points.
After nearly half a century
of supercomputing,
I knew one or two things
about supercomputers
that nobody else knew.
That new knowledge
that I alone had
was how to solve the toughest problems
arising in physics
and how to solve them **at once**.
My mathematical maturity
grew from the algebraic representation
of the Second Law of Motion
of physics
that was written as
F=ma,
or **F**orce equals
mass times acceleration.

That iconic formula, F=ma,
is the most important formula
in mathematics, science, and engineering.
I learned that formula, F=ma,
back in June 1970
and I learned it in the eighth grade
of Christ the King College,
Onitsha, East Central State, Nigeria.

16.2.2 Contributions of Philip Emeagwali to Mathematical Physics

Fast forward a dozen years,
I had taken as my second home
half a dozen mathematics
and physics departments
that were **across** the United States.
My second home
was mathematics and physics departments
from Corvallis (Oregon)
to Washington (District of Columbia)

to College Park (Maryland).
For my first sixteen years
in the United States,
I made those mathematics
and physics departments
my second home.
After sixteen years
in mathematics and physics departments,
I had grown intellectually
to become a research
computational mathematician
and to become a research
computational physicist.
That intellectual maturity
as a research mathematician
and as a research physicist
was what enabled me
to discover thirty-six [36] critical errors
in the advanced textbooks
in the mathematical physics
that governs
the flow of crude oil, injected water,
and natural gas

that flows a mile-deep
inside production oilfields.
I corrected those thirty-six [36]
critical mathematical errors
by inventing
thirty-six [36] partial derivative terms
that ensured that F=ma
and that ensured that
the equality holds
within a system of nine new
partial differential equations
that I invented
and that encoded
the Second Law of Motion
of physics
and that encoded that law
for three-dimensional, three-phase flows
within porous media,
such as flow of crude oil, injected water,
and natural gas **across**
the Oloibiri Oil Field
of Bayelsa State of southeastern Nigeria
that was the first oilfield

that was discovered
in West Africa back in 1956.
Those thirty-six [36]
partial derivative terms encoded
the thirty-six [36]
temporal and spatial components
of the inertial forces
in the three phases
of crude oil, injected water, and natural gas
that were flowing along
the temporal
and the three spatial directions
of my new nine
partial differential equations
of modern calculus.
I continued **across**
my ensemble of 64 binary thousand
tightly-coupled,
commodity-off-the-shelf processors
that shared nothing with each other
and I continued and I invented
my new nine partial differential equations.
I discretized the latter

from its formulation
as a differential problem
of modern calculus
to the extreme-scaled problem
of modern algebra.
I did so to enable me
to more accurately encode
the invisible subterranean motion
of water
that is pumped into water injection wells
and pumped because
water is cheaper than oil and natural gas
and is, therefore, used to push
oil and natural gas
towards production wells.

16.2.3 Things I Did in the Nation's Capital

For me, the answer
to the Grand Challenge Problem
of mathematics and supercomputing
demanded my sixteen-year-long journey,
onward of June 20, 1974

from Corvallis, Oregon, United States
through Los Alamos, New Mexico,
United States
to receiving the top prize in supercomputing
from Silicon Valley, California.
A research pure mathematician
is re-searching a new proof
of a deep and abstract theorem.
I sojourned
to an unknown world of technology.
In that sojourn, I used
sixteen supercomputer-hopefuls
as my guideposts along the new pathways
to the never-before-seen
global network of processors
that is a new internet
that is a new supercomputer
and that is a new computer.
A research applied mathematician
is seeking
a new and better mathematical description,
such as a system of coupled, non-linear,
time-dependent, and state-of-the-art

partial differential equations
that governs
an initial-boundary value problem
of modern calculus.
A research computational mathematician
is seeking
a faster computational model,
that explains or predicts
a phenomenon
that is otherwise impossible
to explain or predict.
In the late 1970s and early ‘80s,
I was a research
computational mathematician
that lived in room eight hundred
and seventy-seven [877]
of Meridian Hill Hall
at twenty-six hundred and one [2601]
16th Street
that was at the corner of
16th Street and Euclid Street
of Adams-Morgan neighborhood
of northwest Washington, D.C.

On a Friday evening, I might have dinner
with my future-wife, Dale.
Our favorite place
was an Ethiopian restaurant
that was a short walk
down 18th street
from Columbia Road
in the Adams-Morgan neighborhood.
My other favorite places are the small cafes
along "T" and "U" streets,
and the Meridian Hill Park Drum Circle.
In the late 1970s and early '80s,
I was most likely to be seen
in the nation's capital, Washington, D.C,
jogging **across** the Smithsonian's
National Zoo of Washington, D.C.
at 6 a.m. in the mornings.
I also walked or jogged **across**
the Adams-Morgan neighborhood,
or **across** the Dupont Circle neighborhood,
or **across** the Foggy Bottom neighborhood,
or **across** Rock Creek Park.
The 12-mile-long **Rock Creek Park**

extends from the Potomac River
to the border of Maryland.
Foggy Bottom
is named after the fog
that clings to the neighborhood
in the morning.
Foggy Bottom
is a late 18th-century neighborhood
and is one of the oldest neighborhoods
in Washington, D.C.
The Dupont Circle
has a famous traffic circle
and a water fountain at its center.
For me, the Dupont Circle **drum circle**
and its array of late afternoon drummers
and dancers
was like a church.
Fast forward to the late 1990s,
and from the drum circles
of Washington, D.C.
to West African drummers
in Baltimore, Maryland.
My wife Dale and I took lessons

on African dance and **rhythm**
and from a charismatic griot
and choreographer
named **Kibibi Ajanku.**
Kibibi Ajanku
was the founding mother
of the **Sankofa Dance Theatre**
of Baltimore, Maryland.

16.2.4 My Night Life in the Nation's Capital

From October 1978 to May 1981,
I lived a short walk
from the Kilimanjaro Restaurant
and Nightclub
that was started in 1982.
Kilimanjaro Restaurant
and Nightclub was throbbing with
live African music and reggae.
Kilimanjaro Restaurant
and Nightclub
was in the Adams-Morgan neighborhood
of Washington, D.C.

In the early 1980s,
I lived at 1915
East-West Highway,
apartment three zero three [303],
Silver Spring, Maryland.
And I walked the short distance
to the railway subway station
called Metro Station.
That underground railway station
was at the boundary
between Silver Spring, Maryland
and Washington, District of Columbia.
On weekdays, I took the
Washington, D.C. rapid transit system
to Foggy Bottom Station
that is west of the White House
and in downtown Washington.
Foggy Bottom
was a very short walk
to the computer center
where I programmed fast computers
and used them to solve
the most computation-intensive

initial-boundary value problems
of modern calculus
that was the mathematical foundation
of computational fluid dynamics.

16.2.5 My Supercomputing in Washington, DC

In the early 1980s,
my massively parallel processing
supercomputer research interests
were multidisciplinary.
I investigated
computation-intensive problems
arising in mathematical biology,
such as the evolution equations.
I investigated
fluid-structure interactions
within the cardiovascular system.
And I investigated
hydrodynamical computations
that included
the simulation of the propagation
of tsunami waves

arising from earthquakes.
I was an unorthodox
research supercomputer scientist.
The mathematician’s blackboard
that is scribbled with matrices,
tensors, summation indices,
and partial derivatives
is not the playground
of a traditional computer scientist.
A traditional computer scientist
that challenged me
on the mathematician’s blackboard
was like **Daniel**
challenging the lion in the lion’s den.

16.2.6 A Day in My Life

By the late afternoons of the early 1980s,
I become mentally exhausted
from programming computers
to solve extreme-scaled
partial **difference** equations

of modern algebra
that arose from and approximated
partial **differential** equations
of mathematical physics.
In the late weekday afternoon
of the early 1980s,
I often walked to the nearby
Embassy of Nigeria
to read the *Daily Times* newspaper
and socialize with Embassy staff
and Nigerian visitors.
At about 4:30 p.m.,
I will be playing tennis
at the nearby four lighted tennis courts
that were at the intersection
of “O” Street northwest
and “23rd” street northwest.
If it’s too cold for outdoor tennis,
I will swim for two hours
at the Olympic-sized indoor swimming pool
that was located at
600 22nd Street northwest,
Washington, D.C.

Or I might play indoor tennis
or play squash
in the indoor courts
at the same address.
About one Friday evening a month,
I will spend a few hours
in the nearby swanky Zanzibar Night Club.
Zanzibar
maintained its image
as the classy African
and international night club
of Washington, D.C.
Zanzibar
enforced a strict dress code.
To be admitted into **Zanzibar**,
gentlemen must wear a dignified
national dress,
such as the Nigerian agbada,
or wear a suit
along with a collared shirt,
a tie, and dress shoes.
In the mid-1980s,
Zanzibar Night Club

was then in the
Foggy Bottom neighborhood
of Washington, D.C.

16.2.7 Dairy of a Black Meteorologist

Back in 1981,
I was doing more rigorous
mathematical analyses
than I was programming supercomputers.
I did my mathematical analyses
inside the **Gramax Heliport Building**
in Silver Spring, Maryland.
The **Gramax Heliport Building**
was an approved landing pad
for helicopters.
In the early 1980s,
the **Gramax Heliport Building**
at 8060 13th Street, Silver Spring, Maryland
was the headquarters
of the U.S. National Weather Service.
I was at the National Weather Service
because of my interest

in using the fastest supercomputers
to solve
the primitive equations of meteorology.
The primitive equations
were a system of coupled, non-linear,
time-dependent, and state-of-the-art
partial differential equations
of modern calculus.
The primitive equations
are used to model the flow of air
and moisture **across**
the surface of the Earth.
I was at the U.S. National Weather Service
because the **Gramax Building**
was a walking distance
from the Silver Spring Metro station,
in Silver Spring, Maryland.
The **Gramax Building**
was also a brisk walk
from my residence
at 1915 East-West Highway,
Apartment 303, Silver Spring, Maryland.
A few months earlier

and in early 1981,
I was still living at Room 877,
of Meridian Hill Hall
that was at the corner of
16th and Euclid Streets,
of the Adams-Morgan neighborhood,
of northwest, Washington,
District of Columbia.

16.2.8 My Struggles to Invent New Algebra

Inside the **Gramax Building**,
I wrote and rewrote and re-calculated
a hundred versions
of my Taylor's expansion
of various finite difference discretizations
of the system of nine
coupled, non-linear, time-dependent,
and state-of-the-art
partial differential equations
of modern calculus,
called Philip Emeagwali's Equations
that governed

my initial-boundary value problem
that was at the mathematical foundation
of my computational fluid dynamics
codes.
I invented those nine
partial differential equations
and they are my contributions
to mathematics.
As a research numerical analyst,
my research goal was to invent
nine corresponding
finite **difference** algorithms
or discretization schemes,
and finite **difference**
algebraic approximations
of my nine partial differential equations,
called Philip Emeagwali's Equations.
I discretized
my new partial differential equations
and I did so
in a manner that will enable me
to first
define and email

their algebraic approximations
as well as their companion codes
and email them
as 65,536
petroleum reservoir simulations
or email them as 65,536
extreme-scale
computational physics problems
that made the news headlines
when I invented
how to solve them
in parallel
and solve them **across**
my new global network of
65,536 tightly-coupled,
commodity-off-the-shelf processors.
Each of my 65,536 processors
operated its own operating system.
And each processor
had its own dedicated memory
that shared nothing with each other.
Those identical processors
were married together

by 1,048,576 identical email wires
and married together
as one seamless, cohesive unit
that is a new supercomputer
that encircled the globe
in the way the internet does
in sixteen-dimensional hyperspace.
My invention was not a new computer *per se*
but was a new internet *de facto*.

16.2.9 My Struggles to Invent New Mathematics

My laborious Taylor's expansions
of 1981
were how I approximated
the value of each of my solution
by taking the sum of its derivatives
at a given point.
Taylor's expansions yielded
my *a priori* error estimates
that I used to pre-select
the most, hopefully, accurate
finite difference algebraic approximations

of the nine partial differential equations,
called Philip Emeagwali's Equations,
that I invented.
I contributed to mathematical knowledge
and my contribution to algebra and calculus
was the cover story
of top mathematics publications,
such as the June 1990 issue
of the *SIAM News*.
One fact that I never mentioned before
was that I often pursued
false mathematical trails.
Back in 1981,
I was unreasonably obsessed
with the **Hopscotch algorithms**
as a numerical solution of
partial differential equations.
I was obsessed with **Hopscotch methods**
because I was unreasonably optimistic
and believed that
Hopscotch methods are hybrid
explicit-implicit methods
that could be very accurate

and that **Hopscotch methods**
could enable me to email
my initial-boundary value problems
and email them **across**
a new internet
that I visualized
as a new global network of
65,536 commodity-off-the-shelf processors.
After a year of seemingly endless
mathematical analyses
of **Hopscotch algorithms**
and computational experiments
of **Hopscotch**
computational fluid dynamics codes
I discovered that
I was following a false trail
and that **hopscotch algorithms**
were over hyped.
After wasting extraordinary amount of time,
I resettled
on explicit finite difference approximations.
In the end,

I invented explicit finite difference
algebraic approximations
of the nine partial differential equations,
called Philip Emeagwali's Equations,
that I contributed to modern calculus.
That was how I scribbled new calculus
that had never been scribbled
on any blackboard before.
That was how I coded new algebra
that had never been coded
by any computational algebraist before.
That was how I saw a new supercomputer
that had never been seen
by any supercomputer scientist before.

16.2.10 Father of Large-Scale Algebra

It's often said that
parallel processing **across**
millions upon millions
of tightly-coupled commodity-off-the-shelf processors
that shared nothing with each other

is the biggest advance in computing
since the programmable computer
was invented
back in 1946.
In my country of birth, Nigeria,
a million billion trillion
floating-point arithmetical computations
are massively parallel processed
each day
and massively parallel processed
to discover and recover
the otherwise elusive

crude oil and natural gas
that are buried a mile deep
in the Niger-Delta oilfields of Nigeria.
As a discoverer-hopeful, back in 1974,
in Corvallis, Oregon, United States,
I asked a big question
that had never been answered before.
That overarching question was:
“How do we parallel process
across a new internet

that is a new global network of
64 binary thousand computers?”
If that big question
that I asked in 1974
was already answered,
or if parallel processing
was already discovered,
my invention
of the massively parallel processing
supercomputer
will not have been cover stories
and would not have been recorded
in the June 20, 1990 issue
of *The Wall Street Journal*.
If the answer to that big
overarching question
was known,
I would not have gotten telephone calls
from the likes of Steve Jobs
who wanted to know
how I invented
the massively parallel processing
supercomputer

that is faster than
the vector processing supercomputer.
Steve Jobs wanted to know how I recorded
3.1 billion calculations per second.
As an aside, my invention
of parallel processing
that occurred on the Fourth of July 1989
inspired Steve Jobs
to use four processors
that processed in parallel
to also attain a speed of 3.1 billion
calculations per second
and record that speed
in his first Apple personal supercomputers,
called the Power Mac G4.
Steve Jobs introduced
his personal supercomputer
at the Seybold conference
that took place in San Francisco
on August 31, 1999.
Like the modern supercomputer,
the fastest speed in your computer
are coming from parallel processing.

The new supercomputer knowledge
that made the news headlines
was that I—**Philip Emeagwali**—had invented
how to massively parallel process
and that I invented the technology
that drives the modern supercomputer
and invented the technology
on the Fourth of July 1989
and invented the technology
in Los Alamos, New Mexico, United States.
I invented
the parallel processing supercomputer
technology
to enable me to solve
the toughest problems
arising in extreme-scale algebra.
Such mathematical physics problems arise
when trying to discover and recover
crude oil and natural gas
and do so from the Niger-Delta oilfields
of my country of birth, Nigeria.
My quest for the new algebra
that is my contribution to algebra

began with the arithmetic times table
that I memorized in 1960
in first grade
at Saint Patrick's Primary School, Sapele,
in the then Western Region
of the then British West African colony
of Nigeria.
That times table
went to only twelve times twelve.
That times table
was near the beginning of knowledge
of arithmetic.
On the Fourth of July 1989, in Los Alamos,
New Mexico, United States,
I—**Philip Emeagwali**—mathematically
invented
how to massively parallel process
arithmetic times tables
and parallel process them **across**
a new internet
that is a new global network of processors.
I invented
new algorithms, or new instructions,

that told each processor
what to compute within itself
and what to communicate to its up to sixteen
nearest-neighboring processors.
Since the first programmable supercomputer
was invented in 1946,
each supercomputer manufactured
was faithful to its primary mission, namely,
to solve the most extreme-scale problems
arising in computational physics
and to increase productivity,
reduce time-to-solution,
and reduce time-to-market.
Supercomputing is mathematics-intensive.
For that reason,
most supercomputer scientists are, in part,
research computational mathematicians.
In supercomputing
and in computational physics, to discover
is to make the impossible-to-solve
possible-to-solve.
The first person, or the discoverer,
makes the impossible possible,

and thereafter, everybody knows that
parallel processing
is no longer a waste of everybody's time.
I—**Philip Emeagwali**—was credited
for making the invention
of massively parallel processing,
the technology that makes supercomputers
fastest.
I invented
parallel processing
when the supercomputer technology
was scorned, ridiculed, and rejected
by the likes of **Steve Jobs.**
I invented
parallel processing
when the supercomputer technology
was presumed
to be untestable and even wrong.
My discovery
that the impossible-to-solve
arising in extreme-scale
algebraic computations
is possible-to-solve

across a new internet
that is a new supercomputer
and a new computer
was recorded in the June 20, 1990 issue
of the *Wall Street Journal*.

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**.