Massively Parallel Systems

and

Global Optimization

by

Dr. Narendra Karmarkar

narendrakarmarkar@yahoo.com

1. Introduction to Global Optimisation:

Traditionally, large body of research in optimization has focused primarily on "convex" optimization problems that have essentially unique global optimum.

However, in the total space of optimization problems, vast majority of situations fall outside this well understood category. Problems with **multiple, global minima,** isolated from each other, (but obviously of equal value) are quite common. They naturally lead to a number of different types of questions, relating to "direct" problems, "inverse" problems and mixed problems, as described below:

a) **Finding Optimal Solution:** Given a function – which we will refer to as "energy" function – find the minima.

   This means finding the minimum **value** as well as **locations**.

b) **Proving Optimality:** Given one such solution, how does one prove that there is no better solution? (without enumerating them all) What types of proofs are possible? How long such proofs might be? How do different global minima relate to each other?

   To take an example from logic, suppose we take a random 3-SAT formula with small number of variables and appropriate number of clauses, it can easily have millions of satisfying solutions, but if we take just a random collection of million proposed assignments over the same number of variables, can they be precisely the satisfying assignments of any 3-SAT problem with same number of clauses? The point is that although the number of solutions is large, they are highly co-related. Similarly, when a continuous optimization problem has a large number of global minima, they are co-related with each other.

In the context of non-satisfiable formula, are there proofs of non-satifiability that don't enumerate a large number of "partial" assignments?

c) **Inverse Problem:** If we know all (or many) global minima, how can we find the energy function? This is like "reverse engineering" the nature.

d) **Synthesis:** If we are given a specification of desired global minima in terms of their value, locations or shape of energy landscape, how do we design a system with appropriate energy function, possibly subject to further constraints? This is a mixture of "direct" and "inverse" problems in the same context.

Needless to say, all these questions are deeply inter-related to each other and also to how systems occurring in nature behave or how artificially engineered systems might be designed.

2. Optimisation in Physical Systems:

Nature solves optimization problems all the time, since laws of nature are nothing but optimality conditions. They are often expressed in terms of **minimum energy principle.**

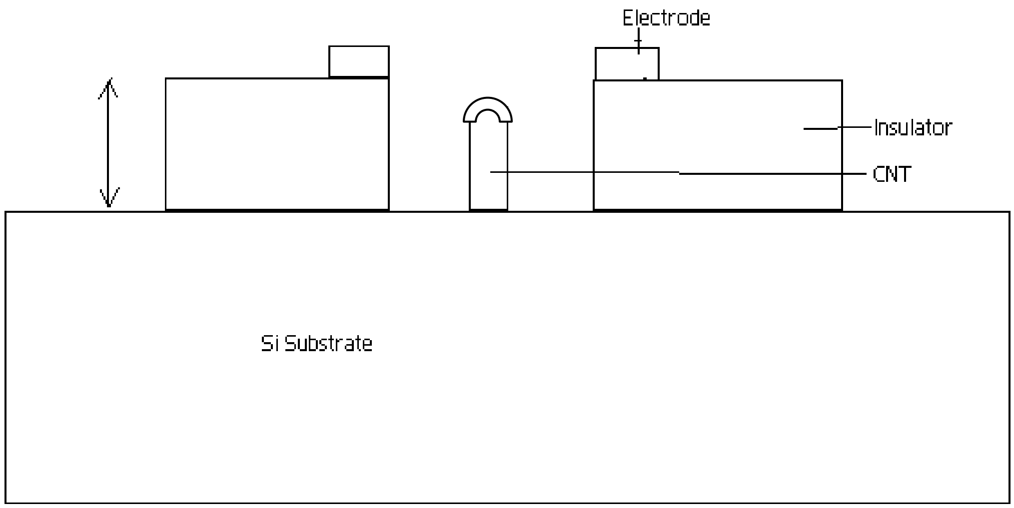
In **path integral formulation** of quantum mechanics, one imagines all possible histories or paths leading to a given state, happening in a massively parallel fashion. This includes even possibilities forbidden classically, called “tunneling”.

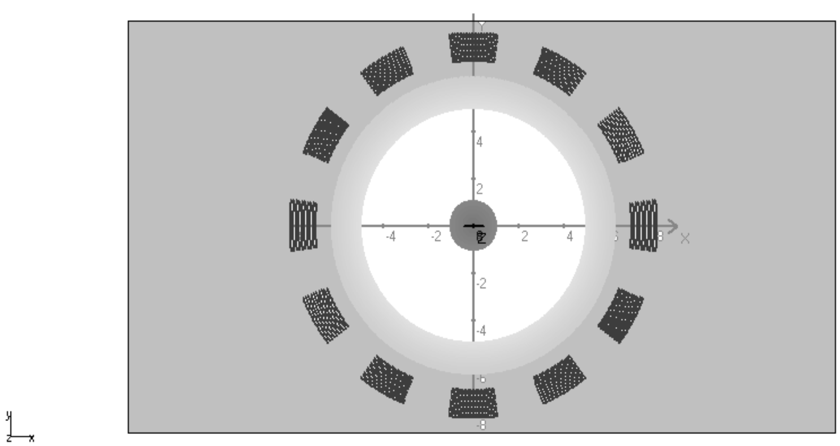
Most of these parallel possibilities interfere destructively due to complex phase associated with the action – only paths close to minimum of the action integral interfere constructively to add up to appreciable probability.

However, the minimum may not be unique; there can be multiple isolated global optima.

Let us take a simple example of field emission from a graphen molecule. Application of electric field induces electrons to tunnel out. While the Fowler–Nordheim theory [Ref #] assumes a one dimensional barrier that is “flat” in the other two dimensions, actual situation in this example is more complex and the probability of tunneling as a function of angle has six distinct global minima of equal value. This has been observed experimentally by many researchers [Ref #]

Here is a simple device to exploit this for communicating information with high data rate; which using a carbon nanotube instead of grapheme, but field emission takes place essentially from the tip again with similar symmetric pattern.





Top view

Figure 1 & 2

The electrodes at the top surface serve two functions:

a. To apply electric field to increase tunneling probability and induce field emission from the tip of the carbon nano-tube.

b. By applying different voltages to electrodes as “tie-breaker”, we can disturb the six-fold symmetry so that the action integral along one of the six stationary paths wins. In this way, the field emitted electrons can be used to **encode** one out of six possibilities.

As the tunneling time itself is very small (femtoseconds), the bit rate is limited by other factors – how fast you can change the electrode voltages. If one applies a. c. signal to a pair of antipodal electrodes to encode one out of three possibilities, one can go to much higher frequencies. For example one can use surface plasmons to deliver the control signal to alter tunneling probabilities.

The encoding then induces oscillations in the pattern of emitted electrons in one of the three planes as shown below:

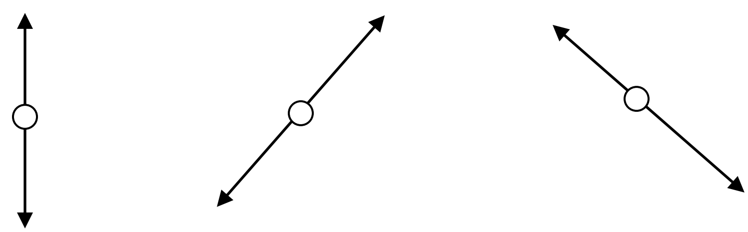


Figure 3

This is just one communication channel, later we will show how to use such channels in a massively parallel manner and reconfigure them at high speeds using the perfect patterns from projective geometry.

As we have seen, the variational problem associated with the path integral can have number of distinct isolated minima and the number can grow rapidly in case of multi-particle problem. Nature has its own way of dealing with such combinatorial complexity.

We have been developing a fresh approach to understanding optimization problems with multiple global minima and made progress on all types of questions mentioned before – finding, proving, reverse engineering etc. But even before we have answers to these problems that we would regard as satisfactory, the insights gained are already turning out to be valuable to a number of new designs of tremendous economic value.

These include:

a. Physical design of the projective geometry machine using massively parallel quantum tunneling that can totally overcome obstacles of latency and bandwidth faced by contemporary designs. The new design can broaden the applicability of massive multi-threading to large and very general classes of computational problems, and can be implemented using already known fabrication techniques.

b. Design of multi-ported, low latency, secondary storage based on magneto-optics, implementing shared memory directly at the physical level, providing a highly valuable feature for data bases and transactional memory.

c. Design of new high bandwidth switches required for next generation internet infrastructure.

d. Design of novel robots with large number of “electro-magnetic fingers” for placing atoms based on complex and sparse patterns of multiple global minima that are more general than regular periodic patterns achieved before using interference lithography.[Ref #]

e. Design of control systems whose stability analysis requires liapunov-like functions with multiple basins of attraction.

f. Design of phased-array radars in terahertz range.

g. Computational calibration of parameters occurring in empirical force fields, whose values may be difficult to measure experimentally but can be reverse engineered from known structure of folded proteins.

3. Multiple Global Minima in Engineered Systems:

3.1 Difficulties associated with Multiple Global Minima:

There are a number of reasons why optimization problems with multiple global minima are so difficult to work with. It was once thought that any algorithms working with such problems will have to deal with exponential number of connected components. However, we have already shown how to circumvent this difficulty. [Contemporary Mathematics v.114]

Even if a level set of such a function is connected, it may be topologically very complex e.g. even in 3-D it may have high genus. We have shown such a body with many “handles” in the figure below:

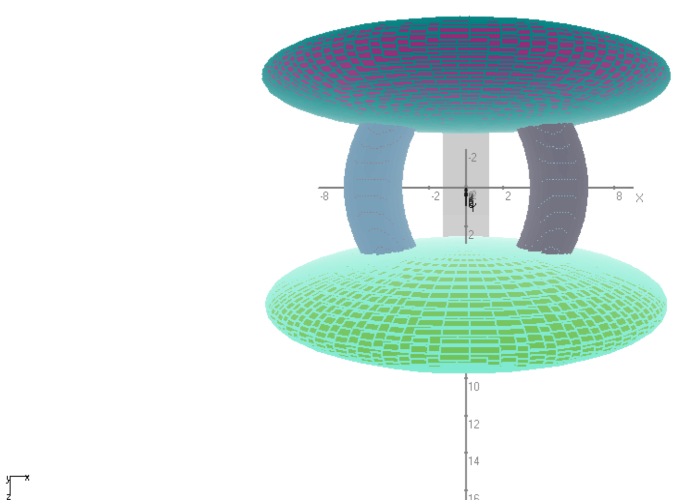


Figure 4 : Level set with many handles

Sculpturing Free Space:

We are going to partition free space into objects like this; they play an important role in synthesizing systems with multiple global minima.

In many naturally occurring systems of this type, there is also a symmetry group relating basins of attraction of different minima.

e.g. one can partition space between two concentric spheres; based on elements of discrete subgroup of SO(3).

Systems based on Electron Optics:

Similar to variational problems associated with path integrals in quantum mechanics, there are action integrals in optics, control theory, electron optics, etc. In this paper we will give examples of problems arising in electron optics.

Electron optics is more complex and offers a rich set of possibilities for new devices for several reasons:

a. It is possible to create curved trajectories relatively easily by setting up appropriate electromagnetic fields.

b. It is relatively easier to interface them to logic circuits in both directions.

c. Electron holography permits exploitation of wave nature of electrons.

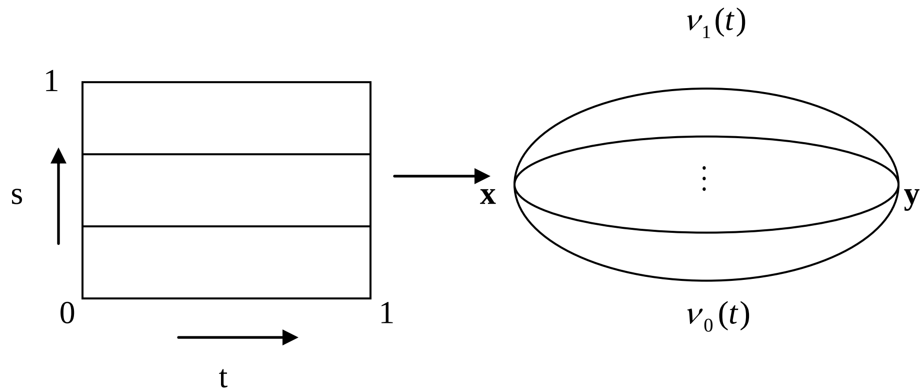
d. Motion of electrons in vacuum is free from collisions. Unlike traditional high-power micro-wave devices, we are interested in ultra-low current and very high performance per watt. Therefore electron density in space is very low and space charge effects can be ignored. In fact, as single-electron transistors become practical they can be used to control field-emission of one electron at a time.

Traditionally, electron optical instruments have been dominated by the concept of “**central optical axis**” in the same way that optimization theory is built around convex problems with unique global optimum. Electrons are required to stay close to central optical axis so that it is easy to analyze and control their behavior. However, once we understand how to deal with optimization problems with multiple global minima, similar insight helps in designing electron optical systems without any dominant central axis.

Electron trajectories in these more general systems can be typically related by **double equivalence** relations:

a) First equivalence relation comes about as a result of lensing action:

   Two electron trajectories $v_0(t)$ and $v_1(t)$, going through the same point **x,** with the same kinetic energy, but traveling in different directions, and converging again in a common point **y,** are considered to be equivalent under lensing action if you can interpolate between the two trajectories by family of trajectories f (s,t) indexed by parameter s, all going through the same point **x** at (t = 0), with same kinetic energy but in different directions and meet again at the common point **y**. This is similar to homotopy relation in differential topology, but with the extra requirement that the electromagnetic fields needed to induce the interpolating electron trajectories are obtainable in source-free vacuum only through application of appropriate boundary conditions



$$f(0,t) = \nu_0(t) \qquad 0 \le t \le 1$$
$$f(1,t) = \nu_1(t) \qquad 0 \le t \le 1$$

$$f(s,0) = \mathbf{x} \quad \text{, \textbf{the starting point}}, \quad 0 \le s \le 1$$
$$f(s,1) = \mathbf{y} \quad \text{, \textbf{the converging point}}, \quad 0 \le s \le 1$$

Trajectory Equivalence based on Lensing Action

Figure 6

The class of trajectories under this equivalence relation belongs to the same partition or fundamental domain of free space.

The second equivalence relation relates different fundamental regions by action of a discrete symmetry group. The action of group elements is also defined on individual electron trajectories to give other valid trajectories. Since the partitioned vacuum through which electron move is itself source-free, we achieve the desired symmetry by expressing the sources on the boundary in terms "symmetrised" multi-poles. For implementing the finite projective geometry architecture, the symmetry group is chosen as an appropriate sub-group of the automorphism group of projective geometry. Groups corresponding to other Cayley graphs can also be chosen, but there are many further advantages to organizing the parallel machine using projective geometry.

For sculpturing the vacuum as described above, we have to solve a new type of boundary value problem. In the usual type of boundary value problem, values of the field on the surface, bounding certain volume are specified and we seek extension of field values to the entire volume subject to satisfying some differential equation.

In the new type of problem we need to solve, a **partition** of the boundary is specified instead of field **values** on the boundary. This may be done in a variety of ways, e.g. it may be in the form of a voronoi diagram or a generalized voronoi diagram based on non-euclidian metric, or based on level sets of a function or partition based on fundamental domains corresponding to a symmetry group or "tiling" of the surface.

We then seek an extension of the boundary partition to a partition of entire bounded volume, which then gets divided into tubes. Electrons move through these tubes so that probability of tunneling in transverse direction between adjacent tubes is negligible compared to tunneling in the longitudinal direction at the end of the tubes. Such division of space is a "soft" partition corresponding to a "perfect pattern" based on the projective geometry [ref. #] and can be easily and rapidly changed or reconfigured simply by changing the boundary conditions.

We describe the physical design of a parallel system based on these ideas in the next section and the projective geometry ideas in the following section.

4. Electromagnetic Cavity Machine:

The central core of this parallel system can be best described as an Electromagnetic Cavity of a very special kind.

The **surface** of this cavity is lined with active logic circuits fabricated with certain modest extension of the current VLSI technology, which is basically a planar process. In the immediate future, this can be based on silicon and might be changed to other possibilities such as graphene in the long run. At the top level, these logic circuits may be organized as millions of ultra low power cores designed to enable massive multithreading. A peta-flop configuration requires only about a square meter of silicon real estate for computing circuits even if they are designed to run at slower clock speed to significantly improve performance per watt. Besides computing, the surface circuits also provide for two types of communication devices between computing resources. The first type is for the traditional 2D nearest neighbor communication.[ref #] The second type provides supporting devices for a new type of **surface normal** communication based on massively parallel quantum tunneling and free space electron optics through the cavity volume. This type of global communication uses highly symmetric flow patterns derived from mathematical structure of finite projective geometry. Its implementation involves a novel electron optical system that does not have any dominant "central optical axis". Instead, it is based on action integrals having multiple global minima. The electromagnetic fields to guide electrons along the required massively parallel trajectories is set up by creating appropriate boundary conditions on the surface of the electromagnetic cavity. The electrodes required to apply such boundary conditions are patterned on the surface of the cavity by standard lithographic techniques. The drivers, receivers and other controlling electronics required for this purpose is located on the surface of the electromagnetic cavity. It is fabricated using standard VLSI process. These supporting devices for the surface-normal communication are described in section -7.

The revolutionary bandwidth and latency properties of our parallel architecture are resulting from combination of perfect patterns of the projective geometry, novel electron optical system and massively parallel surface normal quantum tunneling.

Several other novel devices mentioned earlier are also Electromagnetic Cavity Machines at a conceptual level. According to the end objective, they differ in details such as choice of wavelength or time scales, or what they rearrange - bits of material or bits of information.

5. Brief review of projective geometry architecture:

Most large computational problems contain plenty of parallelism “in principle”. As a result promise of parallelism has long been recognized. However, in practice, the power of parallelism remains grossly under-utilized due to programming difficulties. A good architecture and physical design of a parallel machine should be able to deliver decent efficiency on a wide variety of applications expressed using different programming paradigms. e.g. it should not matter whether the program is written in fortran or lisp or prolog, whether it is doing number crunching, symbolic computation or processing relational data-base queries etc.

One should also strive to provide support for strong scaling. i.e. one should not be required to increase problem size just to show good efficiency with large number of processors. Unless we set the goals or criteria for success high enough, it is unlikely that we can arrive at good architectural solution. It is with such goals in mind that we have devised an architectural scheme based on mathematical properties of projective geometry. In this scheme, it is not necessary to depend upon hand-crafted decomposition of computational problems into parallel tasks. Instead, the hardware has built in rules to automate this to a substantial degree.

This emphasis on wider applicability, strong scalability and reduced programming complexity more than compensates for moderately higher complexity of our physical design. After all, one can manufacture millions of identical copies of hardware, once designed. In contrast, each software package involves a long-drawn evolutionary effort, involving difficult combination of creativity and software discipline. This has been the primary reason why power of computer science remains grossly under-utilized today.

The configuration of a projective geometry is specified by three integers: characteristics of the underlying finite field, **p,** degree of the extension **k,** and dimension of the geometry, **d**. We denote the projective geometry corresponding these parameters by

$$P^d(GF(p^k))$$

Let Ω_l denote the collection of all projective subspaces of dimension l.

Thus,

Ω_0 : set of all points

Ω_1 : set of all lines

Ω_{d-1} : set of all hyperplanes etc

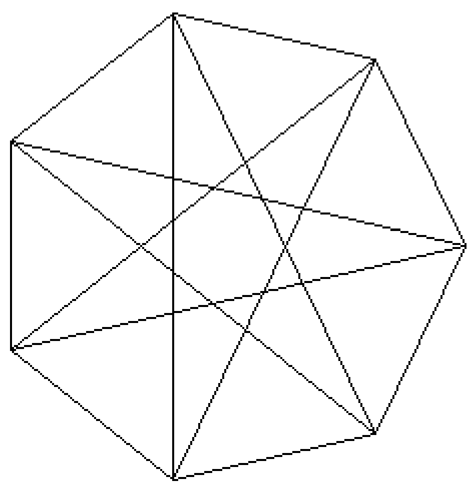
Consider collection of subspaces of dimensions 0, 1, ..$d_{max.}$ In the projective geometry architecture each hardware resource is associated with a subspace and two resources corresponding to subspaces X, Y are connected

iff $X \subseteq Y$

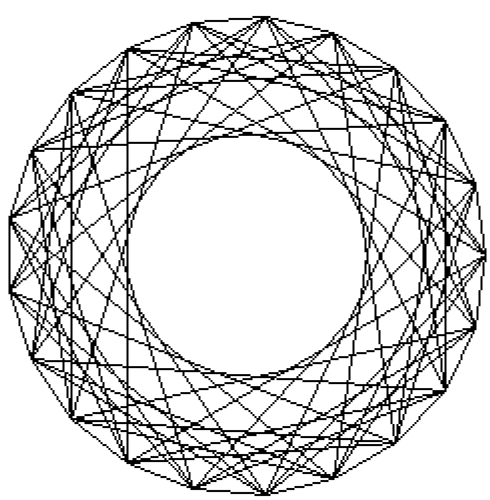
and dim (X) = dim (Y) -1

Here are some examples of 7, 21, 31, 57, 183 point 2d geometries

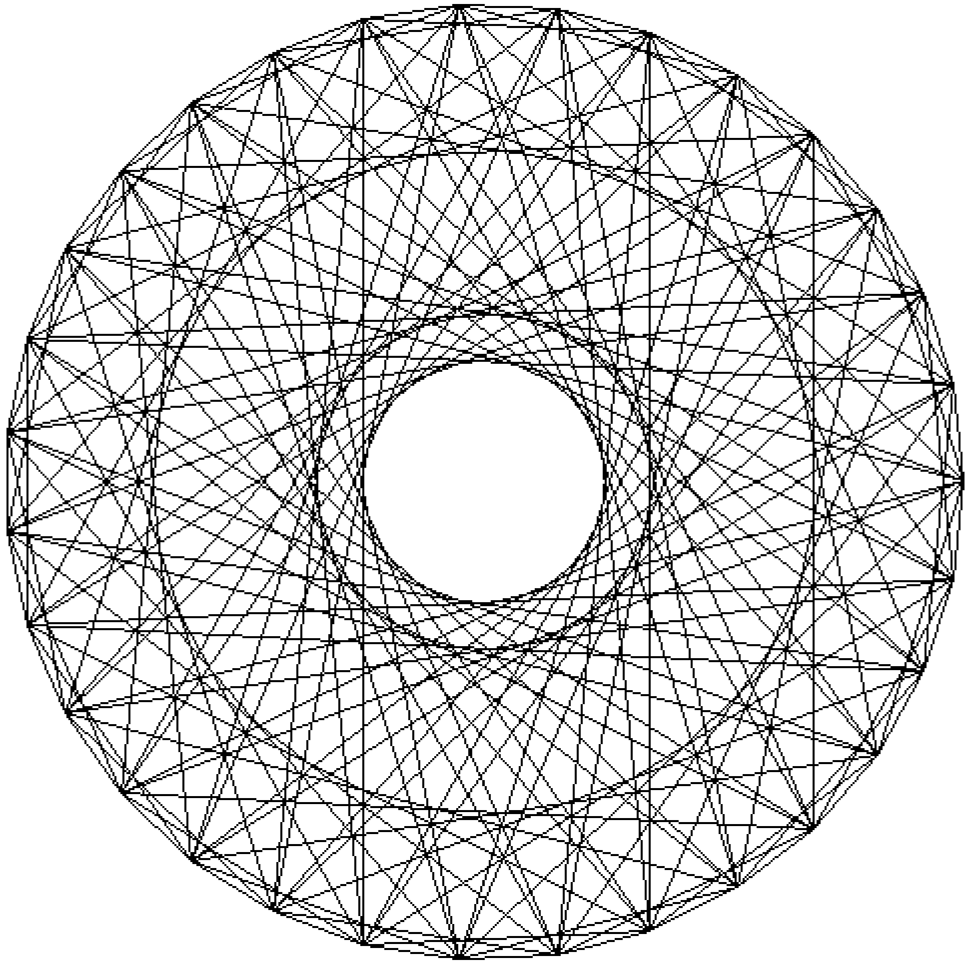
7



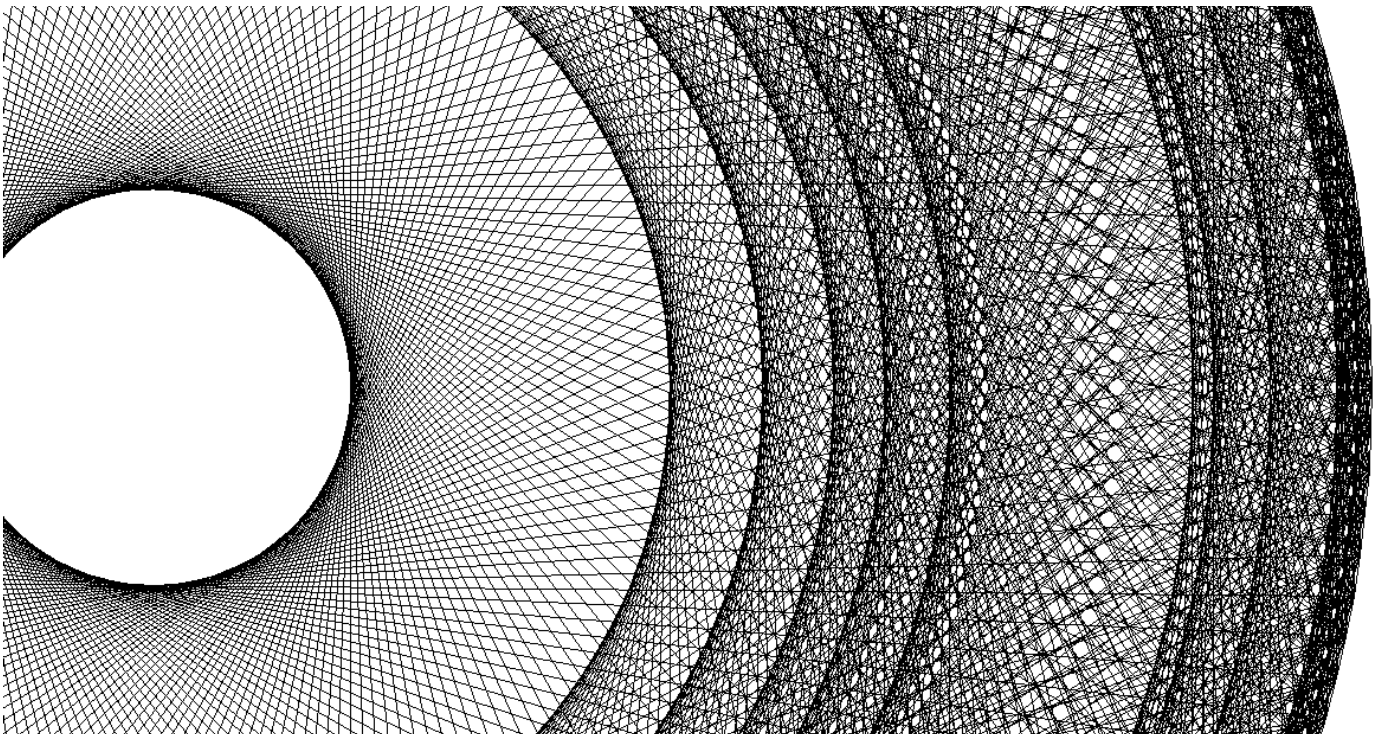
21



31



Portion of 183



The operation of the projective geometry architecture is organized in terms of "**perfect patterns**" as explained in [ref. #]

Perfect Access Pattern for 2-d Geometry is shown in the following table.

Let n = number of points = number of lines

	Point Pairs	Corresponding lines
1	(p1,q1)	l1 = <p1,q1>
2	(p2,q2)	l2 = <p2,q2>
.	.	.
.	.	.
.	.	.
n	(p_n,q_n)	ln = $\langle p_n,q_n \rangle$

Table no 1

A Perfect Access Pattern is a collection of N ordered pairs of points s.t.

1. First members of all pairs (p_1, p_2, … ,p_n) form a permutation of all pts
2. Second members (q_1, q_2, … , q_n) also form a permutation.
3. The lines (l_1, l_2, …, l_n) determined by these pairs form permutations of all lines of the geometry.

Clearly, if one schedules binary operations corresponding to such a set of index – pairs for parallel execution

1. There are no read – write conflicts in memory accesses.
2. There is no conflict in processor usage
3. All processors are fully utilized
4. Memory bandwidth is fully utilized

Furthermore, a collection of perfect patterns is called complete if every index – pair (a,b) occurs in exactly one pattern

Perfect Access Patterns for 4d Geometry is shown in the following table:

Let n = number of planes = number of lines

	Triplet of Points	Triplet of lines			Planes
1	(p_1,q_1,r_1)	$u_1=<p_1,q_1>$	$v_1=<q_1,r_1>$	$w_1=<r_1,p_1>$	$h_1=<p_1,q_1, r_1>$
2	(p_2,q_2,r_2)	$u_2=<p_2,q_2>$	$v_2=<q_2,r_2>$	$w_2=<r_2,p_2>$	$h_2=<p_2,q_2, r_2>$
⋮	⋮	⋮	⋮	⋮	⋮
n	(p_n,q_n,r_n)	$u_n=<p_n,q_n>$	$v_n=<q_n,r_n>$	$w_n=<r_n,p_n>$	$h_n=<p_n,q_n,r_n >$

Table no 2

A **perfect pattern** is a collection of n (non-collinear) triples such that

- Lines u_1, u_2, …, u_n determined by first pair of points from each triplet forms a permutation of all lines.
- Similarly, lines determined by pair (q_i,r_i) form each triplet form a permutation of all lines and lines determined by pairs (r_i, p_i) also form a permutation.
- Planes h_1, h_2, …, h_n determined by the n triplets form a permutation of all planes

A set of perfect patterns is **complete** if every non-collinear triplet occurs in exactly one perfect pattern

Hypothesis graphs:

Organisation of the projective geometry machine in terms of perfect patterns allows exploitation of parallism at a fine grain level. To enable this in presence of conditional branches, we use a concept of hypothesis graphs, which is a slight extension of data flow graphs, augmented with Boolean predicates. A Boolean variable associated with a data variable represents validity of the data value. Operations in the data flow graph compute Boolean predicates in the same way as data operations. When Boolean predicate associated with a node evaluates to false, that branch of the computation terminates. No computation is ever rolled "back".

Although this concept was originally meant to help deal with conditional branches, it is also turning out be extremely useful for efficient computation and simulation under multiple scenarios if they share significant amount of common computation.

e.g. you may start the computation by having Boolean variables c_1, c_2, …,c_n, where c_i represents the condition that price of certain stock is expected to be in the interval $[a_i, b_i]$, or that thickness t of sheet metal while optimizing design of an automobile body is in certain interval $[a_i, a_{i+1}]$. The initial Boolean variables may sometimes persist till the end as symbolic variables or they may get evaluated as a consequence some intermediate calculation.

Virtual Memory Organization based on subspaces

Generally, virtual memory is organized in a hierarchical fashion: Total memory space is divided into pages. Pages consist of words and words consist of bits.

In the projective geometry architecture there are **superpages** organized in the form of lattice. Two superpages are either disjoint or intersect in another superpage (or page at the bottommost level). Each superpage is associated with a subspace of the projective space, and intersection of two superpages is associated with the intersection of corresponding subspaces. Accessing many pages from the same superpage can lead to predictive fetching of the entire superpage enabling exploitation of another kind of locality that is frequently present in many applications.

Disciplined pointers based on memory spaces:

On one hand, pointers enable efficient programs. On the other hand they are also source of many programming errors which are difficult to find. In the projective geometry based organization, we use a concept of "**disciplined pointers**": A pointer variable is associated with a memory subspace of the geometry. It is allowed to point only to target addresses belonging to that subspace. This simple device reduces programming errors.

Application of group theoretic structure of projective geometry:

The symmetries underlying projective geometry play an important role in generating perfect patterns as well as their physical implementation. Generation is explained in this section and physical implementation in the following section:

Observe the following relation between

$$P^d(GF(s)) \quad \text{and} \quad GF(s^{d+1})$$

- $GF(s^{d+1})$ contains a subfield $GF(s)$
- $GF(s^{d+1})$ is a vector space over $GF(s)$ of dimension $d+1$

Hence,

- Non-zero elements of GF(s^{d+1}) $\xleftrightarrow[\text{correspondence}]{\text{one-to-one}}$ Non-zero elements of V^{d+1}(GF(s))

Let G = multiplicative group of non-zero elements of GF(s^{d+1})

H = subgroup of non-zero elements of GF(s)

- Cosets of H in G $\longleftrightarrow$ Rays through origin in V^{d+1} (GF(s))
- Quotient group G/H $\longleftrightarrow$ P^d(GF(s))

The group G is cyclic

Let x = generator i.e. primitive root of GF(s^{d+1})

- Then G = $\{1, x, x^2, \cdots, x^{n+1}\}, \; x^n = 1$

Let $n_d = \dfrac{s^{d+1} - 1}{s - 1}$

- H = $\{1, x^{n_d}, x^{2n_d}, \cdots, x^{(s-2)n_d}\}$

Hence Quotient group G|H can be represented as

- G|H $\longrightarrow$ $\{1, x, x^2, ..., x^{n_d - 1}\}$

We use this method of labeling point in P^d(GF(s))

Represent Points of P^d(GF(s)) as

$$\{1, x, x^2, ..., x^{n-1}\}, \qquad x^n = 1$$

Where n = the number of points.

- The shift operation is a permutation of points

$f : x^i \to x^{i+1}$

$$\sum_j a_j y_j = 0 \;\; \Rightarrow \;\; \sum_j a_j f(y_j) = 0$$

Thus any linear relation is preserved by this operation.

- It maps

  lines $\longrightarrow$ lines

  planes $\longrightarrow$ planes

  Any subspace $\longrightarrow$ Another subspace

  Of dimension k Of dimension k

  Such a mapping is called automorphism of the geometry.

- Furthermore, in a 2-d geometry, by repeated application shift operation you can move

  any point $\longrightarrow$ any other point

  any line $\longrightarrow$ any other line

  i.e group generated by shift operation is transitive on points and lines in $P^2(GF(s))$

  How to generate complete set of perfect patterns for 2d geometry

- To generate a single perfect pattern,

  Take any pair of points a, b a $\neq$ b and apply the shift operation repeatedly

- $(a,b) \longrightarrow$ $l_1 = \langle a,b \rangle$
- (xa, xb) $l_2 = \langle xa, xb \rangle$
- (x^2a, x^2b) $l_3 = \langle x^2a, x^2b \rangle$
- $\vdots$ $\vdots$
- $(x^{n-1}a, x^{n-1}b)$ $l_n = \langle x^{n-1}a, x^{n-1}b \rangle$

To generate a **complete** set of perfect patters, take any line

$l = \{a_1, a_2, \cdots, a_k\}$

form all $\binom{k}{2}$ point pairs from l

$$\left|\begin{matrix}(a_1, a_2)\\ \downarrow\\ shift\end{matrix}\right| \left|\begin{matrix}(a_1, a_2)\\ \downarrow\\ shift\end{matrix}\right| \cdots\cdots \left|\begin{matrix}(a_{k-1}, a_k)\\ \downarrow\\ shift\end{matrix}\right|$$

Each pair gives a perfect pattern by shifting. Together they give a **complete** set

To generate a **complete set** of perfect patterns in a 4-d geometry, We need more than just cyclic shifts.

Other examples of automorphisms

In a finite field of characteristic **p**, the operation

$$x \rightarrow x^p$$

Is an automorphism of the field.

i.e $(x+y)^p = x^p + y^p$

and $(xy)^p = x^p y^p$ for all x,y

We can construct an automorphism of $P^d(GF(s))$, where $s = p^k$,

Using the relationship between

$P^d(GF(s))$ and $GF(s^{d+1})$

$$\sum_j a_{ij} x_j = 0 \quad \Rightarrow \quad \sum_j (a_{ij})^p x_j^p = 0$$

Hence subspaces $\longrightarrow$ subspaces

More general Automorphism:

Let A: any $(d + 1) \times (d + 1)$ nonsingular matrix over $GF(s)$

Consider the mapping

$$x \rightarrow Ax \qquad \text{For} \qquad x \in V^{d+1}(GF(s))$$

This extends to a well-defined map on $P^d(GF(s))$ and gives an automorphism.

Transitivity:
In any P^d ($GF(s)$), the group of all automorphism acts transitively on **all subspaces** i.e. given any pair of subspaces
H_1 and H_2 of same dimension
There is an automorphism that maps,
$H_1 \rightarrow H_2$

Generation of Perfect Pattern for 4-d Geometry:

- Take any non-collinear triplet (a, b, c)
- Apply the elements of automorphism group to generate the **orbit**
  $(a^{(k)}, b^{(k)}, c^{(k)})$ $k = 0, 1, 2, \ldots, n-1$
- The orbit is a perfect pattern

To generate a **complete set** of **perfect patterns**

- Take any plane $H = (x_1, x_2, \ldots, x_k)$
- Form all non-collinear triplets from points of H
- *Generate a perfect pattern/triplet as its orbit*

Together they give a **complete set**

Further packing of patterns, for concepts such as

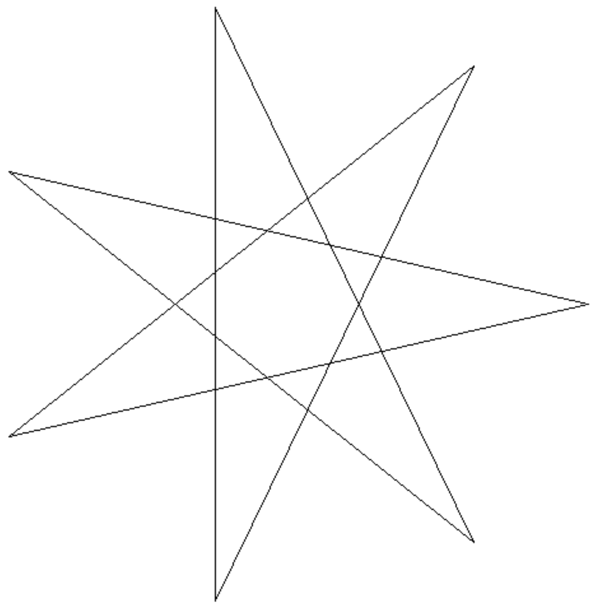
- Complete collection
- *K*-fold complete collection
  - Each pair is covered exactly k times.

And Perfect Sequences of Pattern, refer to [ref #]

6. Communication based on electron optics through the cavity volume:

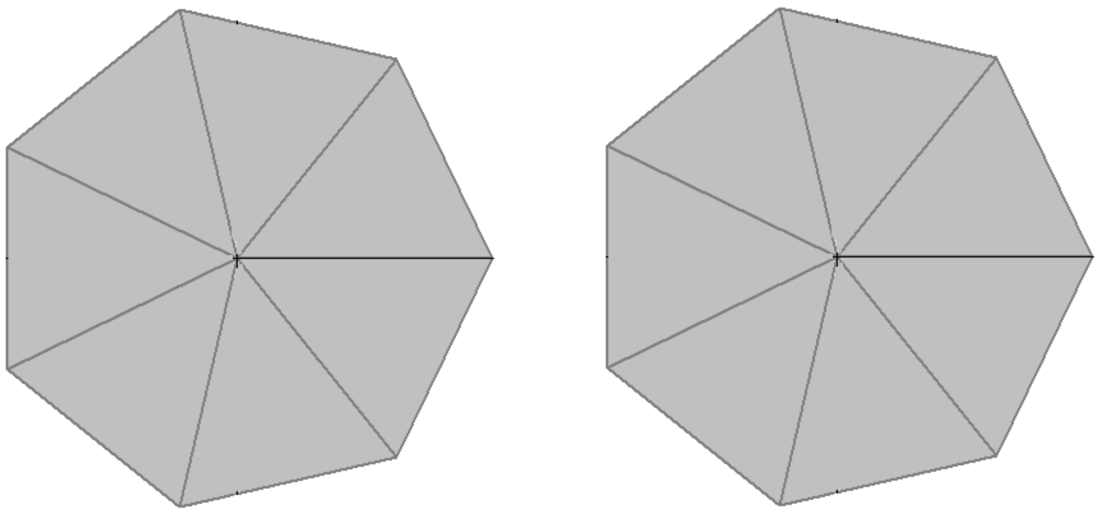
We choose a complete set of perfect patterns or a k-fold covering of perfect patterns [ref. #]. Corresponding to each perfect pattern, the cavity volume is partitioned into fundamental domains. Each fundamental domain contains a “tube” in free space for field-emitted electrons. The ends of the tube have field emitters and detectors located on the cavity surface.

The electromagnetic field required to propel field-emitted electrons along these tubes is created by applying appropriate boundary conditions to the electrodes on the cavity surface. We will illustrate this with a simple example of one perfect pattern for the smallest 7-point projective geometry. The 2D perfect pattern is shown in Fig. 7 below:



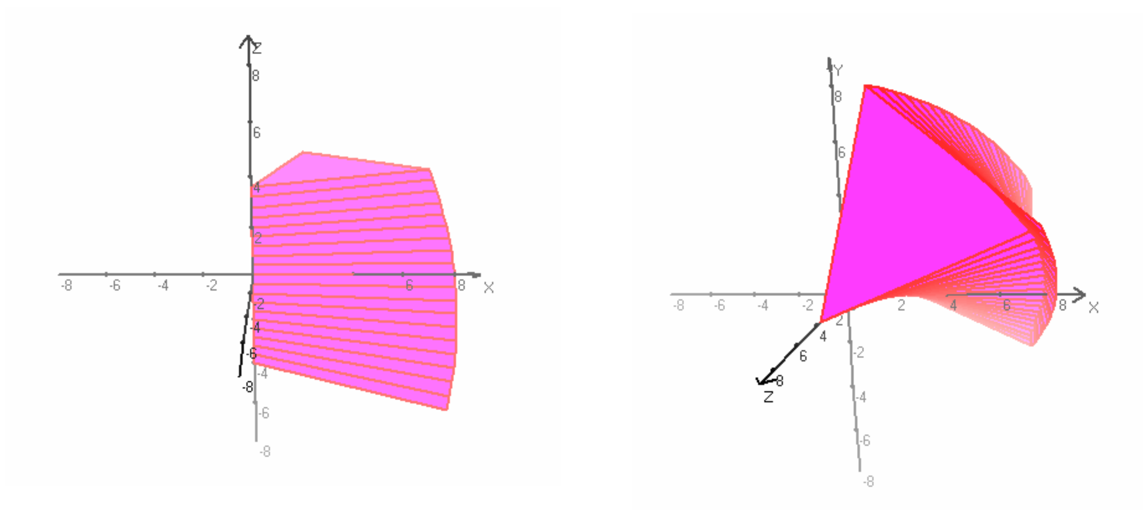
Example of one perfect pattern for 7 point 2d geometry
Fig. 7

The boundary partitions for processors and memories is shown in Fig.8 : They form top and bottom surfaces of the cylindrical cavity in this example.

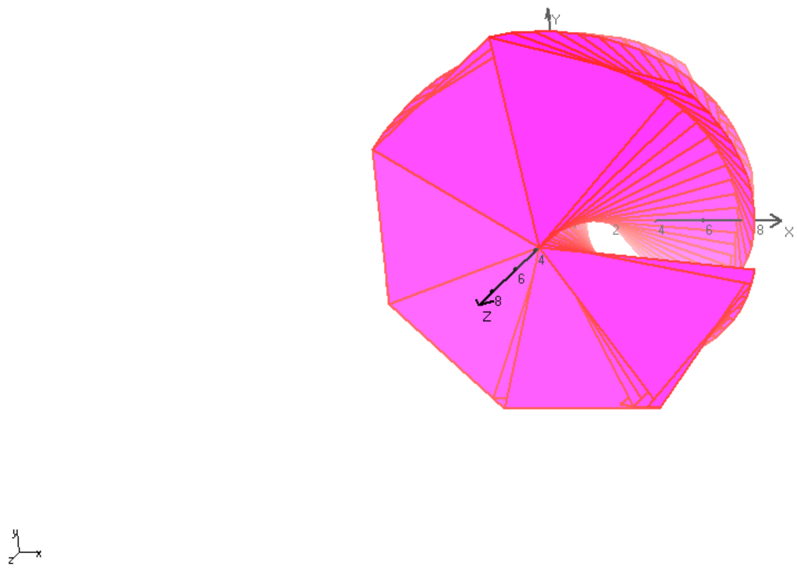


Boundary partition of the top and bottom surface of the cylindrical cavity
Fig

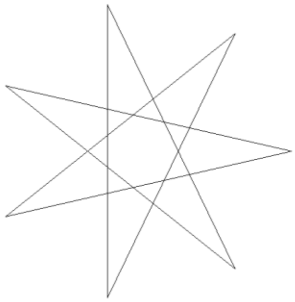
The building block to be used for space partitioning for this example is shown in fig #. The space partition of the cylindrical cavity is shown in Fig. #, the tubes for electron flow are shown in Fig. 9 and fig #:

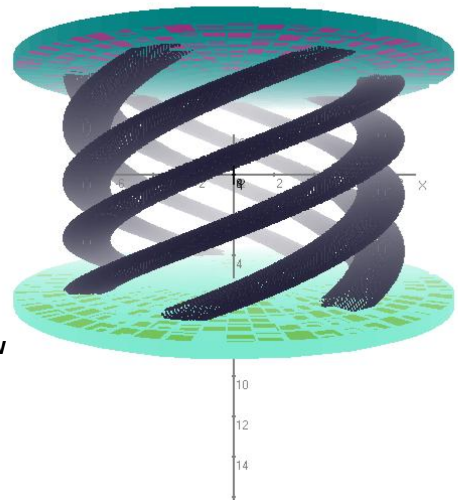


An Example of building block for partitioning free space
Some curved sides of the building blocks have tangents which correspond
To same element of lie algebra
Fig

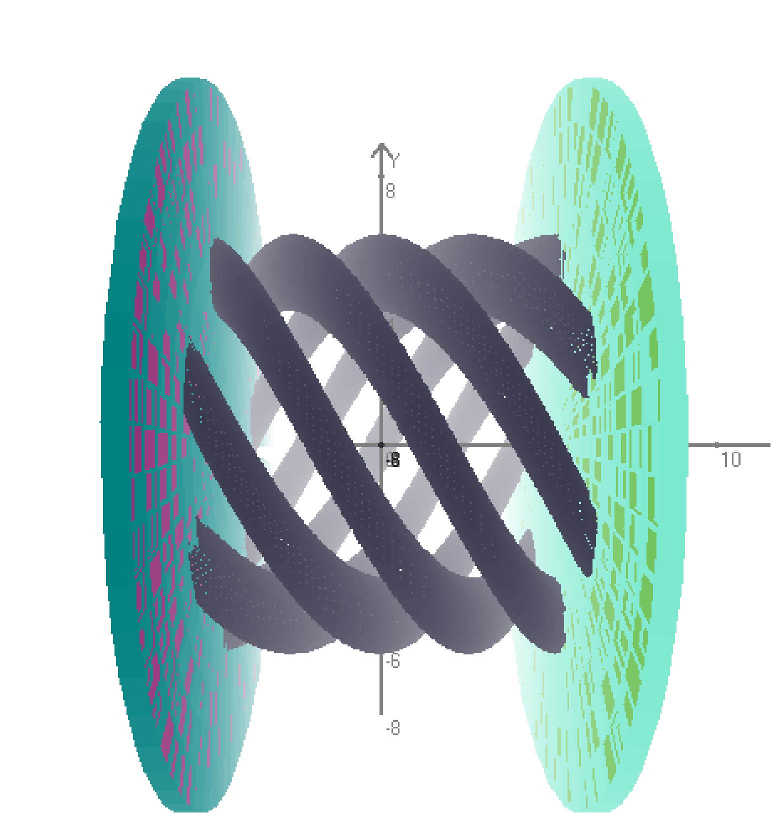


Example of space partition of the cavity volume
Fig. 10



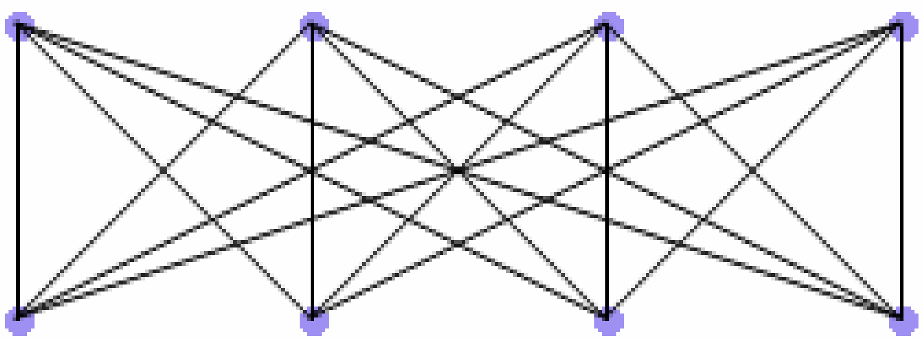


Example of tubes for electron flow
For the perfect pattern above
Fig. 11



The vertical component of the electron motion is due to electric field and the circular component due to magnetic field in the axial direction resulting in overall helical motion. Different perfect patterns from a complete collection can be achieved simply by changing the magnitude of the magnetic field. The physical complexity of connecting n sources to n destinations in one hop using a complete set of perfect patterns is O(n). Alternative arrangements that have been used in contemporary designs for one hop connections is $O(n^2)$.

e.g. The Earth Simulator built in Japan, which is a very good architecture from the point of view of generality, uses connections of $O(n^2)$ physical complexity as shown in Fig. 12 below:



Single hop communication network – complete bipartite graph
Fig. 12

Further refinements of such physical designs have considered replacing electrical cables with optical fibers and using dense WDM multiplexing.[ref #] Such multiplexing saves optical fibers (which is the cheapest component anyway). The total physical complexity of the multiplexers and demultiplexers in the system is still $O(n^2)$, since there are O(n) such components, each of O(n) complexity.

Other methods of reducing interconnect complexity use multiple hops. This approach increases latency, and also multiplies energy used per bit communicated by the number of hops since each time a bit is received, detected, amplified and retransmitted, additional energy is consumed.

In contrast, our design provides low latency, single hop communication channels which can be reconfigured electromagnetically and the physical complexity of the overall design is O(n).

Now we show another way of lining the (approximate) cylindrical cavity. Instead of cylindrical shape, the surface is actually polygonal with many sides. The logic circuits are arranged in a two dimensional pattern. In the figure below we show electron trajectories connecting sources and destination lying on the same vertical line, i.e. they have same angular co-ordinates and each electron trajectory is in meridional plane. In this configuration the magnetic field is only in φ-direction (set up by current along central axis) of the cylinder. It is possible to combine this with helical motion as well.

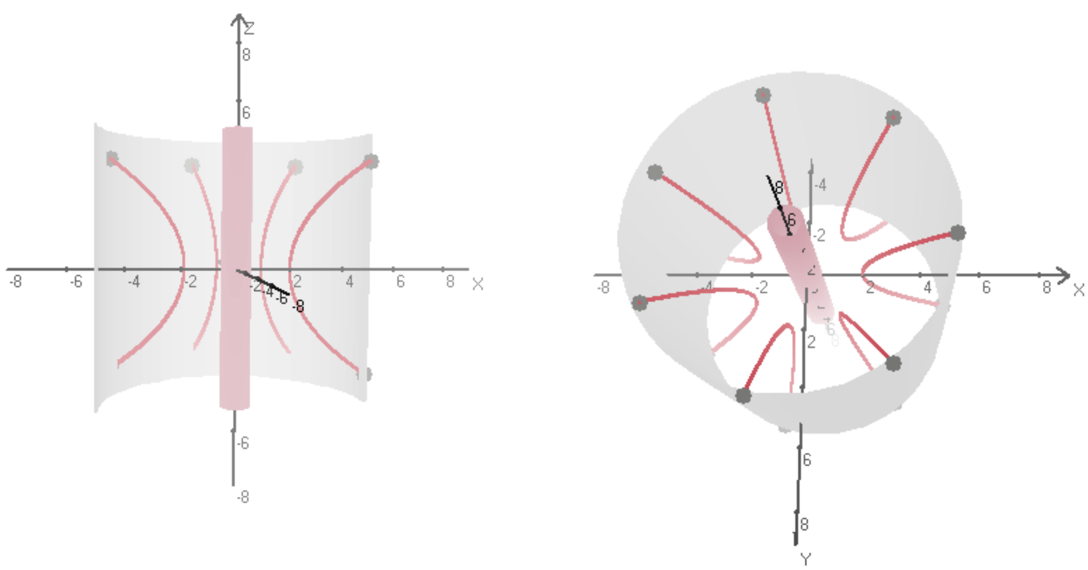
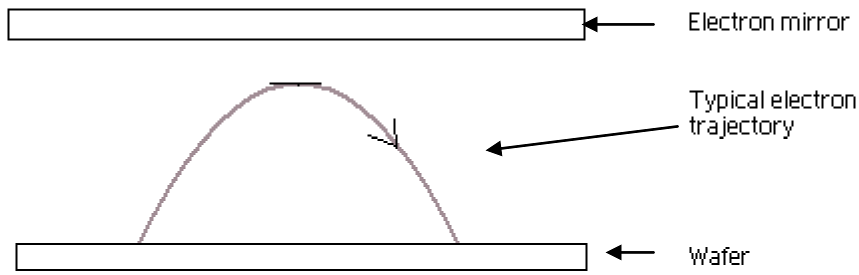


Figure no

There are even simpler geometries possible which are more suitable for single chip or single wafer systems.



Single chip / Wafer Implementation

Figure

In this configuration, an electrode parallel to the wafer surface acts as an “electron mirror”, and emitted electrons are deflected downwards before reaching the top electrode. The deflection electrodes surrounding each emitter control initial direction of launching and thereby select the destination to be reached. Such a design is compact enough to be incorporated in a many core desktop supercomputer.

7. Electromagnetic Cavity Machine- Surface devices

Logic circuits and 2D nearest neighbor communication circuits on the surface of the electromagnetic cavity are standard VLSI devices. The novel devices are for 3D surface normal communication. These consist of arrays of field emitters, detectors, electrodes for extracting, accelerating, modulating, screening, focusing, deflecting and decelerating field-emitted electrons. Before detection, electrons are decelerated to promote “soft landing”, thereby reducing some of damage caused by impact commonly encountered in field emission displays. During deceleration, electrons also return some of energy gained during acceleration, reducing energy dissipation/bit communicated.

There are control circuits for controlling threshold of the onset of field emission, as well as current limiting control circuits to avoid excessive emission. The emitters can be put through different types of cycles or modes – initial start-up mode to clean emitter tips of adsorbed molecules, regular operation mode and periodic “refresh” mode to restore quality of emitter tips and vacuum.

The control for deflection electrodes have two modes differing in time-scales. A slow time constant mode is used for adaptive alignment to compensate for any mechanical misalignment during manufacture or any slowly developing deformation of the cavity due to factors such as slight temperature non-uniformities or structural warping. The fast mode is used for realizing various symmetric communication patterns at run time. Ability to do such finer adjustments or adaptation by fully electronic means is a major practical advantage of electron optics over free-spaces (photon) optics.

For fabricating arrays of field-emission devices, a large number of different approaches have been extensively explored internationally. These include shottky barriers, spindt cathodes made from circular molybdenum cones, sharp pyramidal silicon tips exploiting differential etching rates along different crystal planes, optionally coated with thin films of materials like DLC (diamond like carbon) having negative electron affinity. Arrays of carbon nano-tubes, in particular sparse arrays of single walled CNTs, precisely patterned by means of arrays of dots of Ni or other catalysts seem very promising. For communicating 64-bit words, it is possible to make larger arrays to provide for error correction bits, redundancy to compensate for other types of failure during manufacturing etc.

Unlike optoelectronic devices in the infra-red range, whose packing density is order of magnitude worse than current logic circuits, due to diffraction limit, field emission devices for surface normal communication can be packed with much higher density. In our design, both logic circuits and communication devices on the cavity surface will be ultimately based on quantum tunneling, in tangential and normal directions respectively, and both can be operated as single electron devices.