International Journal of Engineering Sciences Paradigms and Researches (IJESPR)
Volume 48, Issue 03, Quarter 03 (July-Aug-Sept 2019)
An Indexed and Referred Journal with Impact Factor: 2.80
ISSN (Online): 2319-6564
www.ijesonline.com

Implementation of Environmental Setup for Optical Computers

Ashwini Kumar[1] and Devender Kumar Saini[2]

[1]M. Tech. Student,
Department of ECE, GITM, Gurgaon, Haryana, India

[2]Assistant Professor,
Department of ECE, GITM, Gurgaon, Haryana, India

Publishing Date: July 13, 2019

Abstract

An optical computer (also called a photonic computer) is a device that uses the photons in visible light or infrared (IR) beams, rather than electric current, to perform digital computations. An electric current flows at only about 10 percent of the speed of light. This limits the rate at which data can be exchanged over long distances, and is one of the factors that led to the evolution of optical fiber. By applying some of the advantages of visible and/or IR networks at the device and component scale, a computer might someday be developed that can perform operations 10 or more times faster than a conventional electronic computer. Visible-light and IR beams, unlike electric currents, pass through each other without interacting. Several (or many) laser beams can be shone so their paths intersect, but there is no interference among the beams, even when they are confined essentially to two dimensions. Electric currents must be guided around each other, and this makes three-dimensional wiring necessary. Thus, an optical computer, besides being much faster than an electronic one, might also be smaller. Broadly speaking, an optical computer is a computer in which light is used somewhere. This can means fiber optical connections between electronic components, free space connections, or one in which light functions as a mechanism for storage of data, logic or arithmetic. Instead of electrons in silicon integrated circuits, the digital optical computers will be based on photons. Optical technology promises massive upgrades in the efficiency and speed of computers, as well as significant shrinkage in their size and cost. An optical desktop computer could be capable of processing data up to 100,000 times faster than current models because multiple operations can be performed simultaneously.

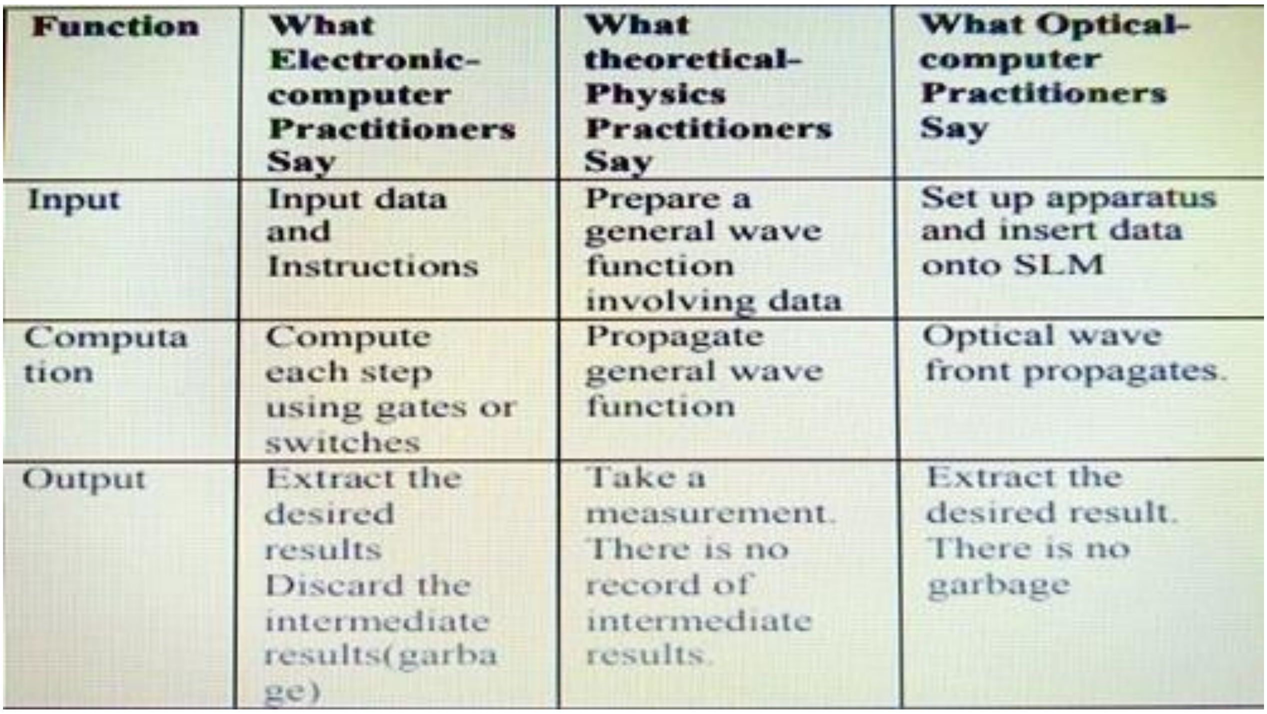
Keywords: *Optical Computers, Environmental Setup, Optical Processors Classical Architectures.*

Introduction

The pressing need for optical technology stems from the fact that today's' computers are limited by the time response of electronic circuits. A solid transmission medium limits both the speed and volume of signals, as well as building up heat that damages components. For example, a one-foot length of wire produces approximately one nanosecond (billionth of a second) of time delay. Extreme miniaturization of tiny electronic components also leads to 'cross-talk' - signal errors that affect the system's reliability. These and other obstacles have led scientists to seek answers in light itself. Light does not have the time response limitations of electronics, does not need insulators, and can even send dozens or hundreds of photon signal streams simultaneously using different color frequencies. Those are immune to electromagnetic interference, and free from electrical short circuits. They have low-loss transmission and provide large bandwidth; i.e. multiplexing capability, capable of communicating several channels in parallel without interference. They are capable of propagating signals within the same or adjacent fibers with essentially no interference or cross talk. They are compact, lightweight, and inexpensive to manufacture, as well as more facile with stored information than magnetic materials. By replacing electrons and wires with photons, fiber optics, crystals, thin films and mirrors, researchers are hoping to build a new generation of computers that work 100 million times faster than today's machines. The fundamental issues associated with optical computing, its advantages over conventional (electronics-based) computing, current applications of optics in computers & problems that remain to be overcome and current research are discussed in this part.

Lasers, fibers, and optical components have already proved their reliability and high levels of performance in many applications such as CD-ROM drives, laser printers, photocopiers and scanners, Storage Area Networks (SANs), optical switches, all-optical data networks, holographic storage devices, and biometric devices at airports to track weapons and drugs. At the same time, the promise of optical computing comes from the many advantages that optical interconnections and optical integrated circuits have over their electronic counterparts. Optical computing is immune to electromagnetic interference and free from electrical short circuits. Photons of different colors can travel together in the same fiber or cross each other in free space without interference or cross-talk. Photons have low-loss transmission and provide large bandwidth, offering multiplexing capacity for communicating several channels in parallel without interference. Optical materials are compact, lightweight, inexpensive to manufacture, more facile with stored information than magnetic materials, and possess superior storage density and accessibility compared to magnetic materials. Progress in holographic storage devices can enable storage of the entire U.S. Library of Congress onto a sugar-cube-size hologram. Furthermore, optical parallel data processing is easier and less expensive than electronic. In addition, optical computing systems offer computational speeds more than 107 times faster than the currently fastest electronic systems. This means a computation that takes a conventional computer more than 11 years to solve would take an optical computer less than one hour. The following table 1 describes different people view about how input, computation and output take place in computing.

Table 1: Different People Opinion about Computing



Function	What Electronic-computer Practitioners Say	What theoretical-Physics Practitioners Say	What Optical-computer Practitioners Say
Input	Input data and Instructions	Prepare a general wave function involving data	Set up apparatus and insert data onto SLM
Computation	Compute each step using gates or switches	Propagate general wave function	Optical wave front propagates.
Output	Extract the desired results Discard the intermediate results(garbage)	Take a measurement. There is no record of intermediate results.	Extract the desired result. There is no garbage

Fundamentals of Optical Information Processing

Optical information processing is based on the idea of using all the properties of speed and parallelism of the light in order to process the information at high-data rate. The information is in the form of an optical signal or image. The inherent parallel processing was often highlighted as one of the key advantages of optical processing compared to electronic processing using computers that are mostly serial. Therefore, optics has an important potential for processing large amount of data in real time.

The Fourier transform property of a lens is the basis of optical computing. When using coherent light, a lens performs in its back focal plane the Fourier transform of a 2D transparency located in its front focal plane. The exact Fourier transform with the amplitude and the phase is computed in an analog way by the lens. All the demonstrations can be found in a book published in 1968 by Goodman and this book is still a reference in the field. The well-known generic architecture of optical processors and the architectures of the optical correlators will be presented successively.

Optical Processors Classical Architectures

At the beginning, real-time pattern recognition was seen as one of the most promising application of optical processors and therefore the following two architectures of optical correlators were proposed.

Figure 1{B} shows the basic correlator called 4-f since the distance between the input plane and the output plane is four times the focal length of the lenses. This very simple architecture is based on the work of Maréchal and Croce in 1953 on spatial filtering and was developed during the following years by several authors.

The input scene is displayed in the input plane which Fourier transform is performed by Lens 1. The complex conjugated of the Fourier transform of the reference is placed in the Fourier plane and therefore multiplied by the Fourier transform of the input scene.

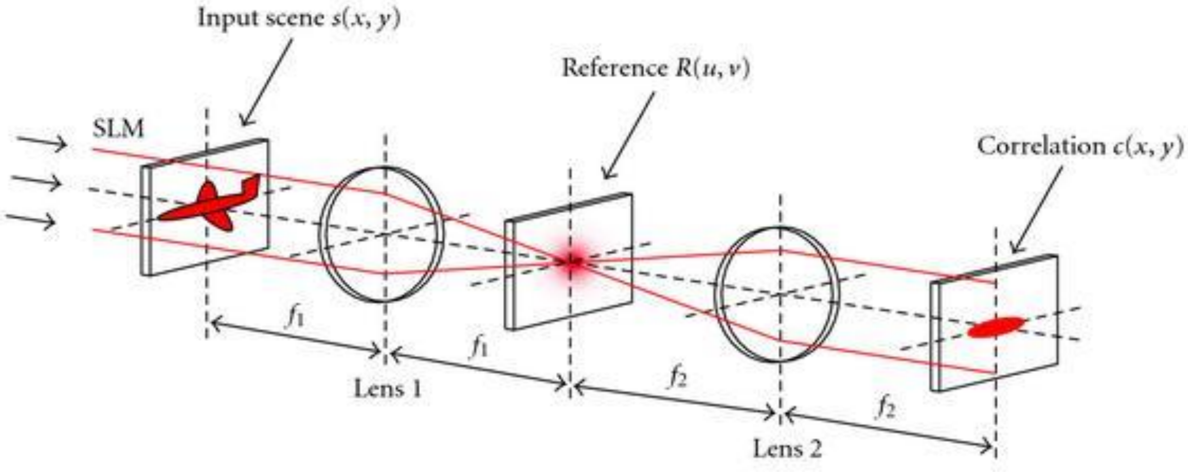


Figure 1{B(a)}

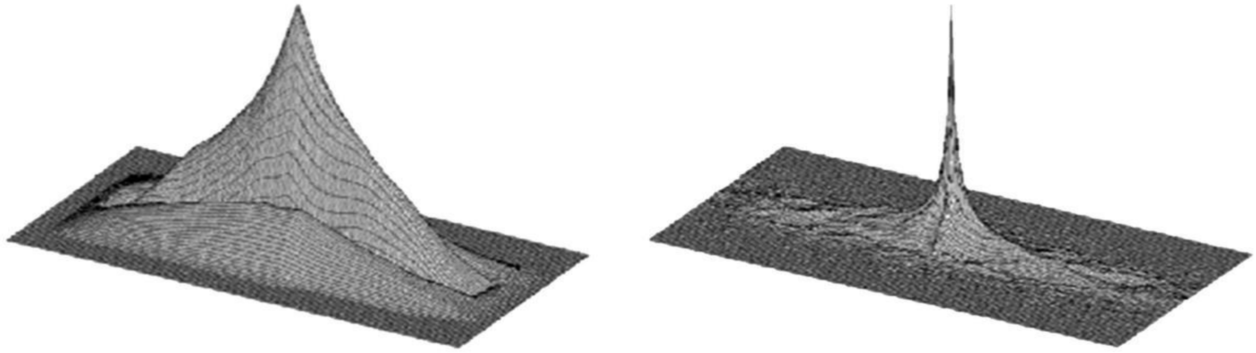


Figure 1{B(b)} **Figure 1{B(c)}**

Figure-1(B): Basic 4-f correlator: (a) Optical Setup. (b) Autocorrelation peak of a matched filter. (c) Autocorrelation peak for a phase only filter

Lens 2 performs a second Fourier transform that gives in the output plane the correlation between the input scene and the reference. Implementing a complex filter with the Fourier transform of the reference was the main challenge of this set-up, and Vander Lugt proposed in 1964 to use a Fourier hologram of the reference as a filter. Figures 1{B(b)} and 1{B(c)} show respectively, the output correlation peak for an autocorrelation when the correlation filter is a matched filter and when it is a phase only filter.

In 1966, Weaver and Goodman presented optical correlator architecture, the joint transform correlator (JTC) that is represented by Figure 1{C}.

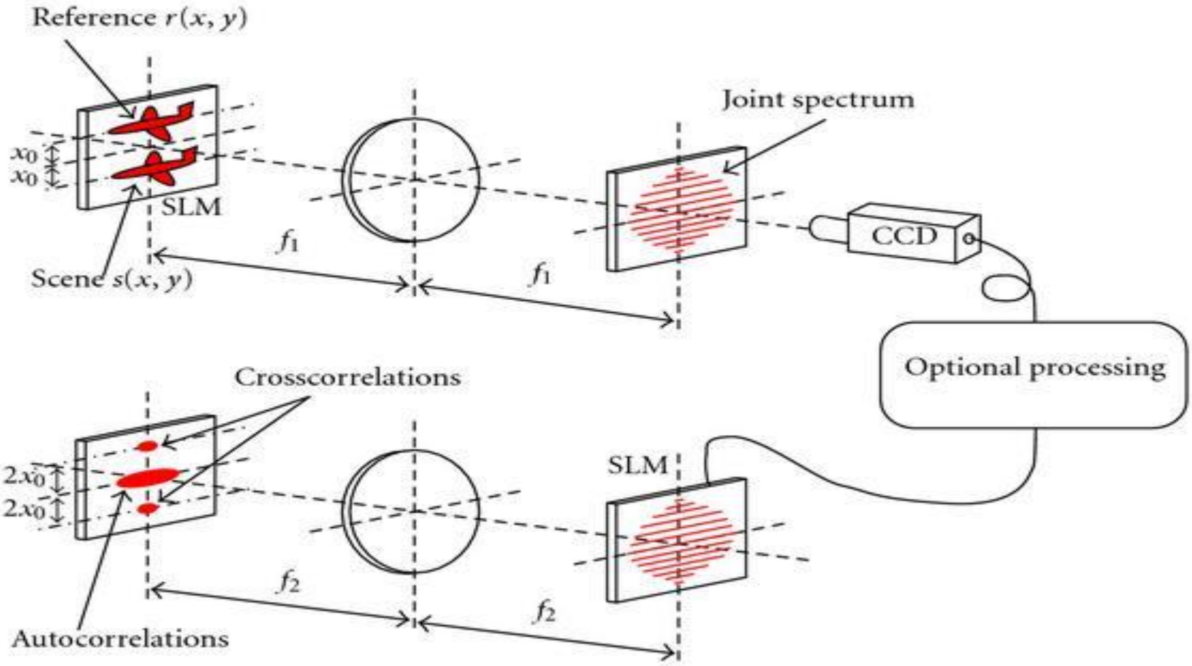


Figure 1{C(a)} - JTC

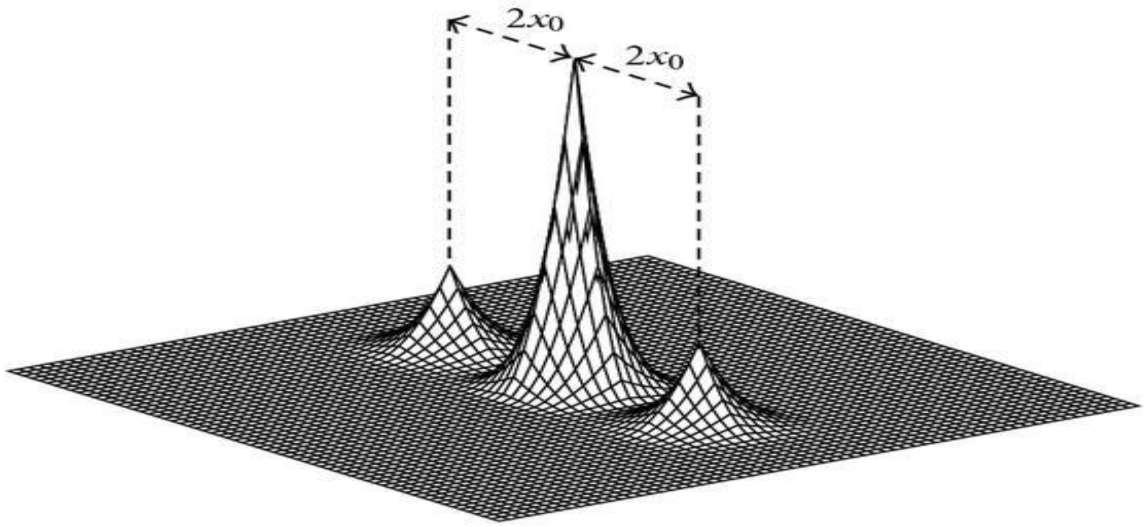


Figure – 1{C(b)}

Figure 1{C}: Joint Transform Correlator (JTC): (a) Optical Setup. (b) Output Plane of JTC

The two images, the reference and the scene are placed side by side in the input plane that is Fourier transformed by the first lens. The intensity of the joint spectrum is detected and then its Fourier transform is performed. This second Fourier transform is composed by several terms including the cross correlations between the scene and the reference. Using a SLM this Fourier transform can be implemented optically as shown on Figure 1{C(a)} Figure 1{C(b)} shows the output plane of the JTC when the reference and the scene are identical. Only the two cross correlation peaks are of interest.

To have a purely optical processor, the CCD camera can be replaced by an optical component such as an optically addressed SLM or a photorefractive crystal. One of the advantages of the JTC is that no correlation filter has to be computed therefore the JTC is the ideal architecture for real-time applications such as target tracking where the reference has to be updated at a high-data rate.

Figures 1{B} and 1{C} represent coherent optical processors. Incoherent optical processors were also proposed: the information is not carried by complex wave amplitudes but by wave intensities. Incoherent processors are not sensitive to the phase variations in the input plane and they exhibit no coherent noise. However, the nonnegative real value of the information imposes to use various tricks for the implementation of some signal processing applications.

Linear optical processing can be decomposed into space-invariant operations such as correlation and convolution or space-variant operations such as coordinates transforms and Hough transform. Nonlinear processing can also be implemented optically such as logarithm transformation, thresholding or analog to digital conversion.

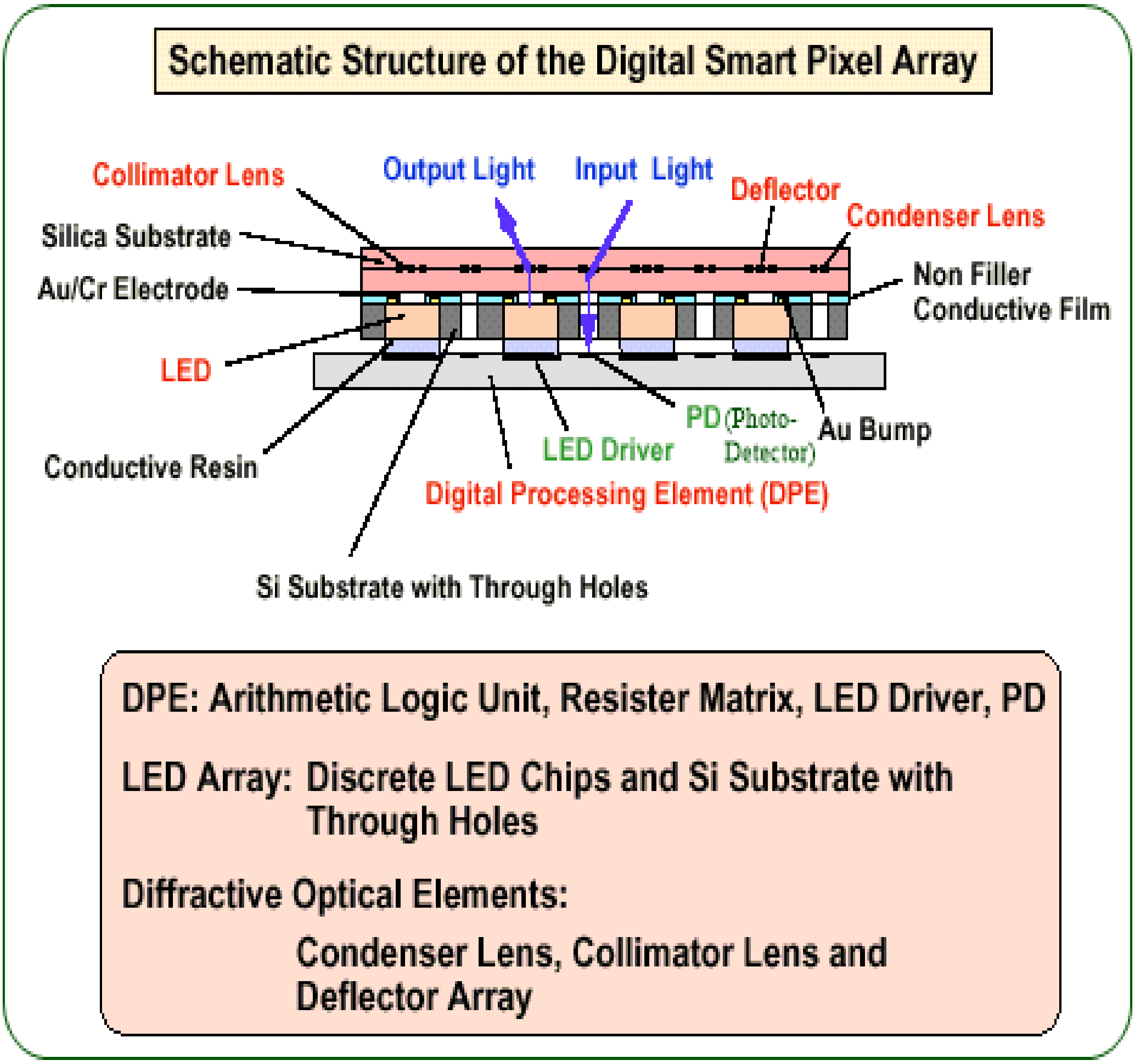


Figure 2{E}: Schematic Structure of the Digital Smart Pixel Array

Fourier Series

A **Fourier Series** is basically a way of approximating or representing a periodic function by a series of **simple harmonic (sine and cosine)** functions.

$$f(x) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx),$$

The corresponding Fourier Coefficients are:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\, dx$$
$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx)\, dx$$
$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx)\, dx$$

And integrations are over a single interval in x of 2L

So the code for Fourier series can be expressed as:

[A0, A, B]=fourier(l,n,f)

Arguments:

l : half of the period, (periodicity of the function f which is to be approximated by Fourier Series).

n: no. of Fourier Coefficients we want to calculate

f: function which is to be approximated by Fourier Series

A0: The first fourier coefficient.

A: An array/matrix whose nth element is the nth coefficient An.

B: An array/matrix whose nth element is the nth coefficient Bn.

The code for scilab is –

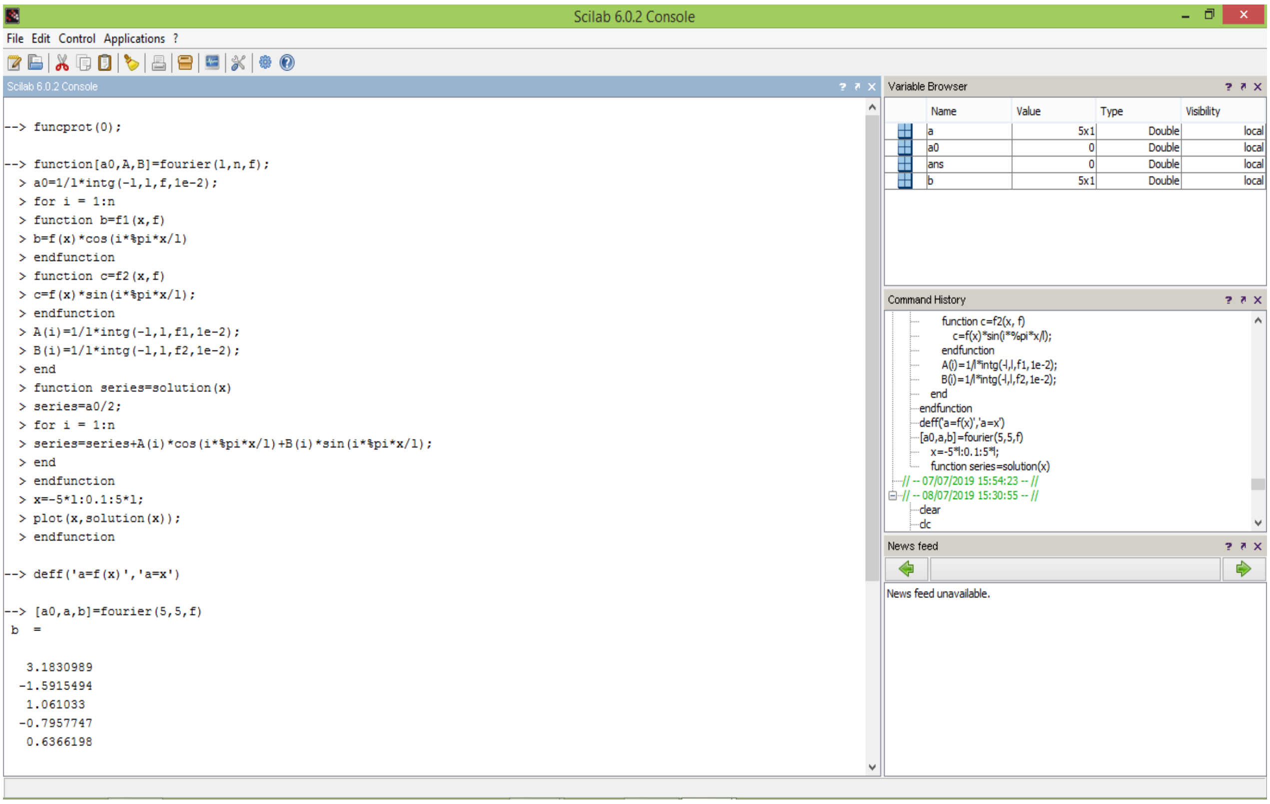
```
//Fourier Series Cefficients
//The following function returns the Fourier Coefficients, ‘An’ & ‘Bn’
//
//We need to provide the following arguments:
//
//l=periodicity of the function ‘f’ which is to be approximated by Fourier Series
//n= no. of Fourier Coefficients we want to calculate
```

```
//f= function which is to be approximated by Fourier Series

funcprot(0);
function[a0,A,B]=fourier(l,n,f);
    a0=1/l*intg(-1,1,f,1e-2);
    for i=1:n
        function b=f1(x,f)
       b=f(x)*cos(i*%pi*x/l)
        endfunction
        function c=f2(x,f)
        c=f(x)*sin(i*%pi*x/l);
        endfunction
        A(i)=1/l*intg(-1,1,f2,1e-2);
        B(i)=1/l*intg(-1,1,f2,1e-2);
    end
    function series=solution(x)
        series=a0/2
        for i=1:n
            series=series+ A(i)* cos(i*%pi*x/l)+B(i)* sin(i*%pi*x/l);
       end
    endfunction
    x=-5*l:0.1:5*l;
    plot(x,solution(x));
endfunction
```

Output:

Deff(‘a=f(x)’,’a=x’)

[a0,a,b]fourier(5,5,f)



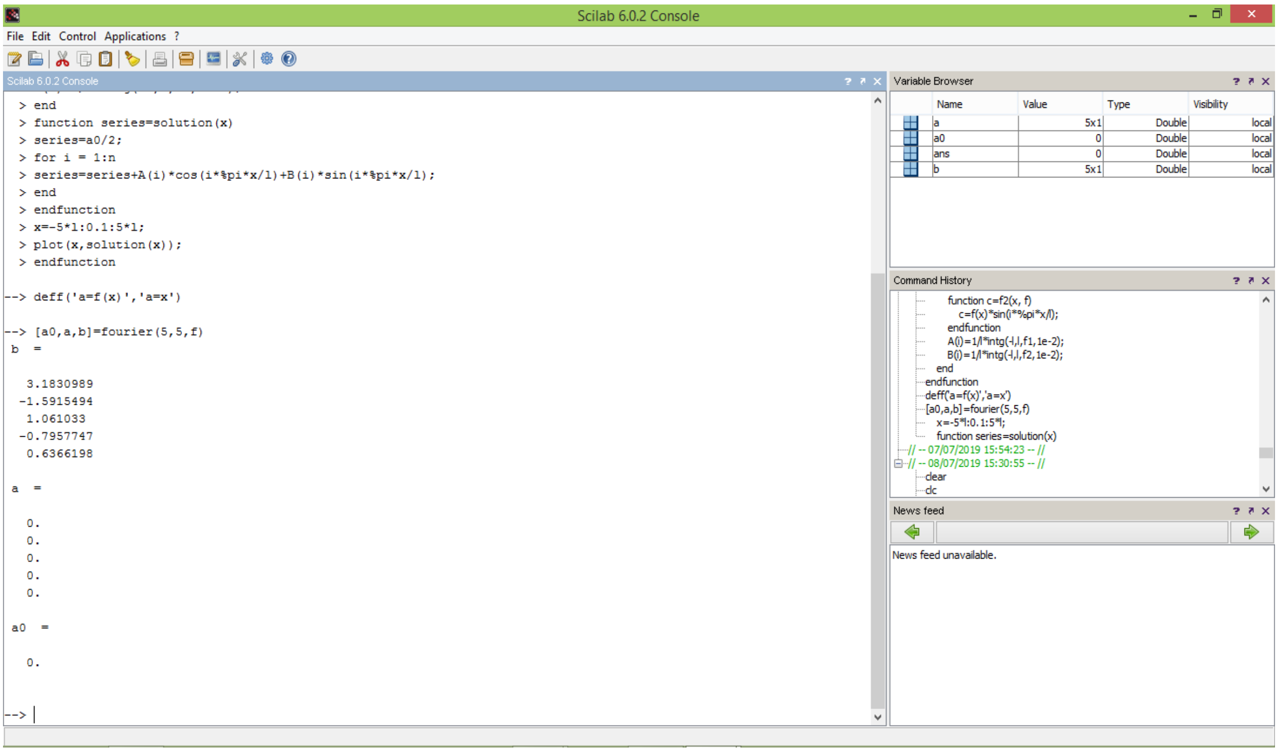
```
--> funcprot(0);

--> function[a0,A,B]=fourier(l,n,f);
  > a0=1/l*intg(-l,l,f,1e-2);
  > for i = 1:n
  > function b=f1(x,f)
  > b=f(x)*cos(i*%pi*x/l)
  > endfunction
  > function c=f2(x,f)
  > c=f(x)*sin(i*%pi*x/l);
  > endfunction
  > A(i)=1/l*intg(-l,l,f1,1e-2);
  > B(i)=1/l*intg(-l,l,f2,1e-2);
  > end
  > function series=solution(x)
  > series=a0/2;
  > for i = 1:n
  > series=series+A(i)*cos(i*%pi*x/l)+B(i)*sin(i*%pi*x/l);
  > end
  > endfunction
  > x=-5*l:0.1:5*l;
  > plot(x,solution(x));
  > endfunction

--> deff('a=f(x)','a=x')

--> [a0,a,b]=fourier(5,5,f)
 b  =

   3.1830989
  -1.5915494
   1.061033
  -0.7957747
   0.6366198
```

Name	Value	Type	Visibility
a	5x1	Double	local
a0	0	Double	local
ans	0	Double	local
b	5x1	Double	local

```
        function c=f2(x, f)
            c=f(x)*sin(i*%pi*x/l);
        endfunction
        A(i)=1/l*intg(-l,l,f1,1e-2);
        B(i)=1/l*intg(-l,l,f2,1e-2);
    end
endfunction
deff('a=f(x)','a=x')
[a0,a,b]=fourier(5,5,f)
    x=-5*l:0.1:5*l;
    function series=solution(x)
// -- 07/07/2019 15:54:23 -- //
// -- 08/07/2019 15:30:55 -- //
    clear
    clc
```

News feed unavailable.

Figure 3{A}: Scilab coding of Fourier Series(Part-1)



```
  > end
  > function series=solution(x)
  > series=a0/2;
  > for i = 1:n
  > series=series+A(i)*cos(i*%pi*x/l)+B(i)*sin(i*%pi*x/l);
  > end
  > endfunction
  > x=-5*l:0.1:5*l;
  > plot(x,solution(x));
  > endfunction

--> deff('a=f(x)','a=x')

--> [a0,a,b]=fourier(5,5,f)
 b  =

   3.1830989
  -1.5915494
   1.061033
  -0.7957747
   0.6366198

 a  =

   0.
   0.
   0.
   0.
   0.

 a0  =

   0.

-->
```

	Name	Value	Type	Visibility
▦	a	5x1	Double	local
▦	a0	0	Double	local
▦	ans	0	Double	local
▦	b	5x1	Double	local

```
        function c=f2(x, f)
          c=f(x)*sin(i*%pi*x/l);
        endfunction
        A(i)=1/l*intg(-l,l,f1,1e-2);
        B(i)=1/l*intg(-l,l,f2,1e-2);
      end
    endfunction
    deff('a=f(x)','a=x')
    [a0,a,b]=fourier(5,5,f)
      x=-5*l:0.1:5*l;
      function series=solution(x)
// -- 07/07/2019 15:54:23 -- //
// -- 08/07/2019 15:30:55 -- //
    clear
    clc
```

News feed unavailable.

Figure 3{B}: Scilab coding of Fourier Series(Part-2)

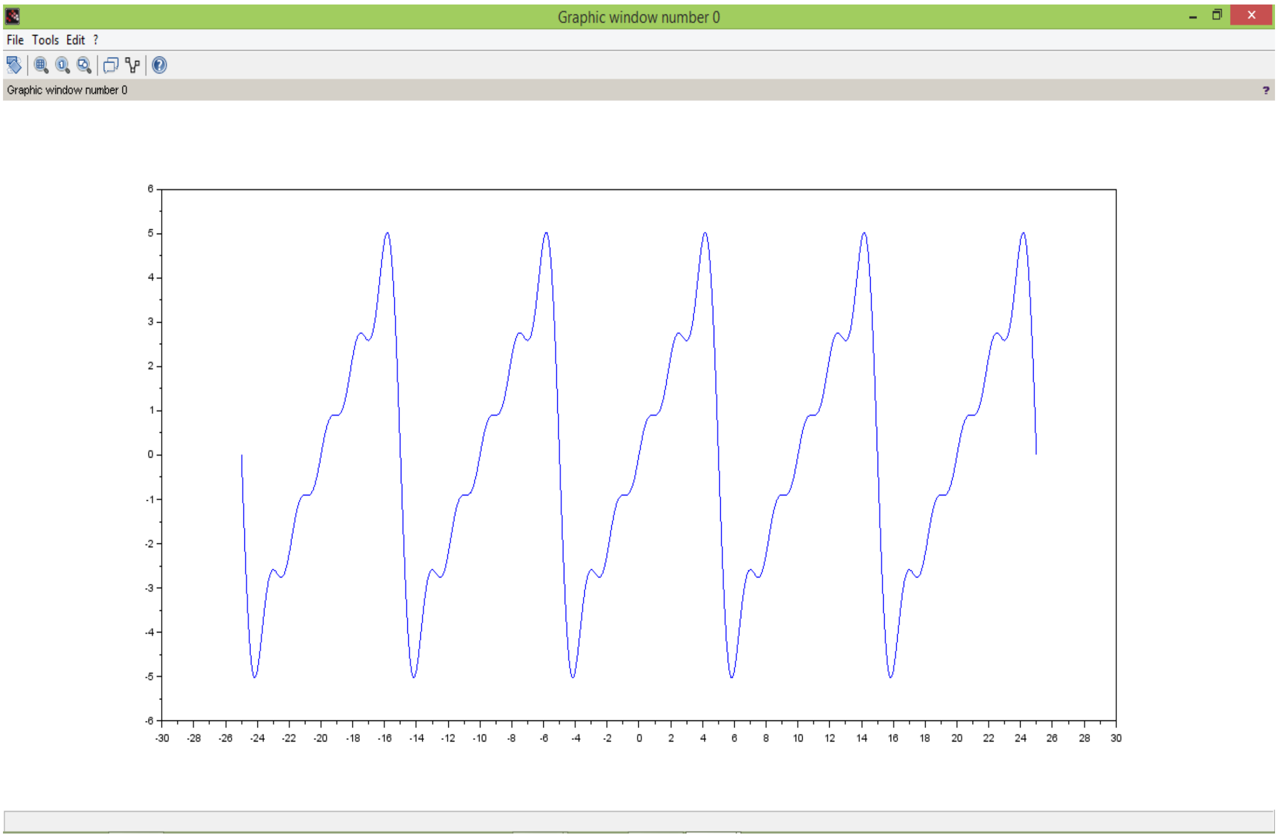
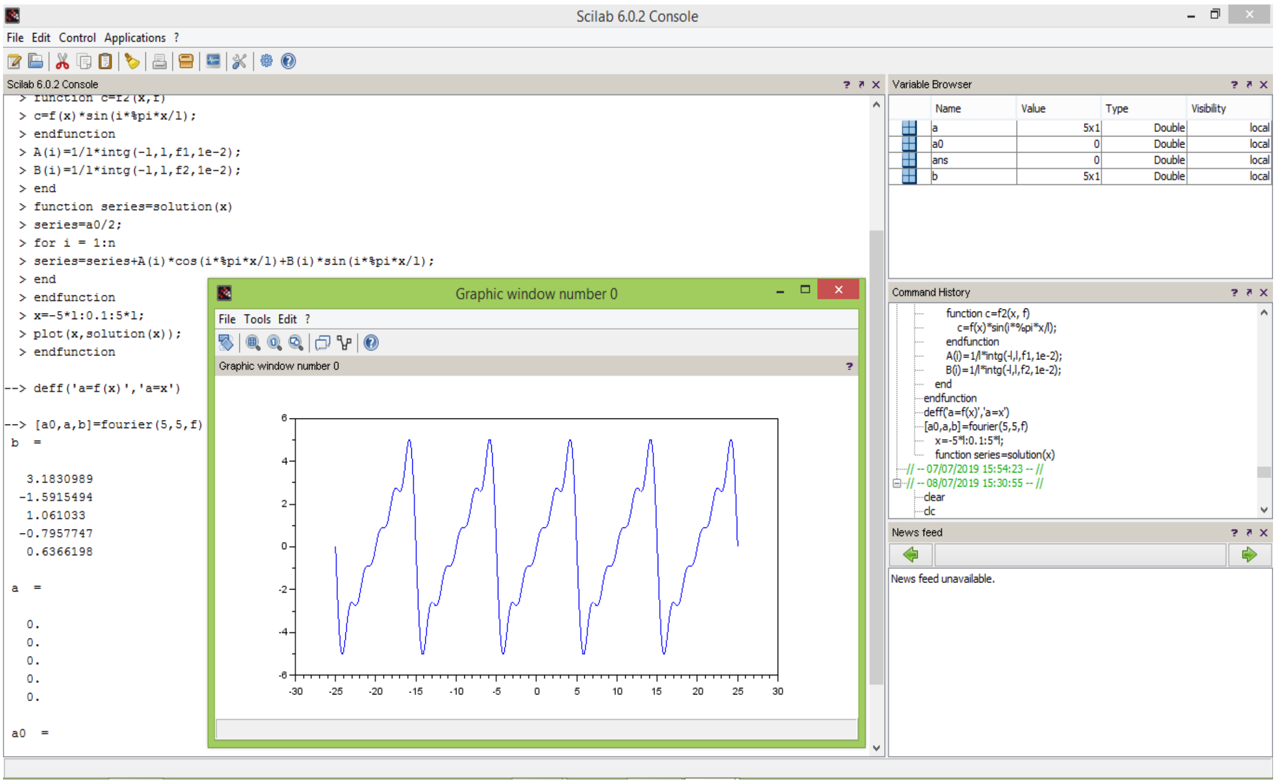


Figure 3{C}: Scilab simulation of Fourier series



```
> function c=f2(x,f)
> c=f(x)*sin(i*%pi*x/l);
> endfunction
> A(i)=1/l*intg(-l,l,f1,1e-2);
> B(i)=1/l*intg(-l,l,f2,1e-2);
> end
> function series=solution(x)
> series=a0/2;
> for i = 1:n
> series=series+A(i)*cos(i*%pi*x/l)+B(i)*sin(i*%pi*x/l);
> end
> endfunction
> x=-5*l:0.1:5*l;
> plot(x,solution(x));
> endfunction

--> deff('a=f(x)','a=x')

--> [a0,a,b]=fourier(5,5,f)
 b  =

   3.1830989
  -1.5915494
   1.061033
  -0.7957747
   0.6366198

 a  =

   0.
   0.
   0.
   0.
   0.

 a0  =
```

	Name	Value	Type	Visibility
	a	5x1	Double	local
	a0	0	Double	local
	ans	0	Double	local
	b	5x1	Double	local

```
        function c=f2(x, f)
           c=f(x)*sin(i*%pi*x/l);
        endfunction
        A(i)=1/l*intg(-l,l,f1,1e-2);
        B(i)=1/l*intg(-l,l,f2,1e-2);
     end
   endfunction
   deff('a=f(x)','a=x')
   [a0,a,b]=fourier(5,5,f)
     x=-5*l:0.1:5*l;
     function series=solution(x)
// -- 07/07/2019 15:54:23 -- //
// -- 08/07/2019 15:30:55 -- //
   clear
   clc
```

News feed unavailable.

Figure 3{D}: Scilab coding & simulation of Fourier series

Conclusion

Research in optical computing has opened up new possibilities in several fields related to high performance computing, high-speed communications. The acceptability of digital optical computing systems as off-the-shelf or dedicated system is still not very high. Optical computing is mostly analogue when electronic computing is digital. The digital optical Computers were not able to compete with the electronic due to the lack of appropriate optical components. Optical processing is useful when the information is optical and that no electronics to optics transducers are needed. Optical computing helps in understanding the architectural problems associated with very high-speed digital computing. Electromagnetic radiation and induction effects are avoided, and experimental demonstrations of both communications and switching at terabit per second bandwidths exist. Current digital architectures are heavily influenced by the assumptions of arbitrary fanout and instantaneous signal propagation within moderately complex subsystems. As switching speeds become faster and power more of a concern, both assumptions prevent architectures from scaling up in speed. This work involves latch-free designs in which finite signal propagation time is fundamental. Such speed scalable designs can take advantage of higher speed devices as they become available. Tools such as the delay distribution algorithms are essential to this style of design. Optics provides an excellent environment in which to study speed of light limited architectures, which are becoming of increasing concern in electronic computer design also. Designing an optical computer involves much more than simply inserting an "optical transistor" into an existing design. The maturity and commercial development of digital electronics suggests that an all-optical computer is not imminent. Optics will probably find its way gradually into digital computers, starting from the fibers already used to connect cabinets in large, high speed systems. Although optical architectures may well be different from electronic ones in important respects, they will probably build on the digital design knowledge base on which electronic computers rest. Optical computers will eventually combine spatial parallelism with high speed design constrained by the speed of light limit, as will future electronic computers. In the meantime, a better understanding of speed of light limited digital systems shows great promise for immediate applications. Communications systems can benefit from even limited optical processing. Time critical tasks in signal processing are another area in which significant applications may exist. Perhaps even more important is the fact that the speed of light limit is a universal phenomenon, not just an optical one. By studying the time-space tradeoffs in the optical domain, insight may be gained into the fundamental nature of physical realizations of the mathematical model which constitutes computation.

References

[1] S. Dolev and M. Oltean (Eds.): OSC 2009, LNCS 5882, pp. 2–4, 2009.

[2] D. H. Hartman, "Digital high speed interconnects: a study of the optical alternative" Opt. Engg. Vol. 25, p1086, 1986.

[3] F. E. Kiamilev, "Performance comparison between opto electronics and VLSI multistage interconnection networks," J. Light. Wave Tech. Vol. 9, p 1674, 1991.

[4] Understanding SONET/ SDH by K Surya Prakash

[5] The ComSoc Guide to Next Generation Optical Transport: SDH/SONET/OTN, by Huub van Helvoort pg no:-5-6

[6] Next Generation SONETISDH, by Stamatios V. Kartalopoulos ISBN 0-471-61530-7 © 2004 by the Institute of Electrical and Electronics Engineers.

[7] By Hossin Abdeldayem and Donald O. Frazier COMMUNICATIONS OF THE ACM September 2007/Vol. 50, No. 9 61

[8] Pierre Ambs Research Article Optical Computing: A 60-Year Adventure pgno:-10

[9] Alexander a.sawchuk, senior member, IEEE, and timothy c.strand, member, IEEE, PROCEEDINGS OF THE IEEE, VOL. 72, NO. 7, JULY 1984

[10] Sawchuk and Strand: Digital Optical Computing, Proceedings of the IEEE. Vol. 72, No. 7, July 1984 Pg: 776

[11] Debabrata Goswami, “Article on optical computing, optical components and storage systems,” Resonance- Journal of science education pp: 56-71 July 2003.