Luminous Flux Computation and the Use of Hybrid Symbolic Integration - Spline Interpolation Method

T. F. Chew[1], A. H. Yahya[2] and H. C. Low[1]

School of Mathematical Sciences
Universiti Sains Malaysia
11800 Minden, Pulau Pinang
Malaysia
([1] hclow@cs.usm.my, [2] ahyahya@cs.usm.my)

Abstract

Lighting calculations are commonly done to evaluate the efficiency of a lighting system. One of the important parameters in establishing a luminaire's efficiency is the total flux emitted from a luminaire or light source. The luminous flux of a light source can be calculated from readily available data of the measurements of its Luminous Intensity Distribution Function, in the form of table of values function. The method currently used in the lighting industry is by means of a semi-manual method of computation and summation of flux at discrete angular steps of luminous intensity under measurement and the use of zonal constants.

In this paper we explore the use of mathematical techniques through software to generate a scheme that provides a more automated and more accurate way of computing the luminous flux. We explore the use of cubic spline interpolation and of symbolic mathematics in a hybrid numeric-analytic integration method, implemented in MATLAB environment.

We also conduct some numerical experiments to verify the efficiency and accuracy of the new method for different simulated light intensity distribution function and compare it to results based on current practices.

1. Introduction

In the field of illumination and display engineering, the quantification of light output is an important activity as it enables the computation of the efficiency of a light source for illumination, indication or display. The quantity for light is luminous flux, with units in lumens, while the quantity for the measure of the concentration of light flux is the luminous intensity.

The relationship between the luminous flux and luminous intensity is given by the flux equation [1]:

$$\phi = \int_{\Omega} I_v d\Omega \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

where I_v is the luminous intensity, in candela

ϕ is the luminous flux, in lumens

Ω is the solid angle subtended in space, in steradians.

By definition the luminous flux is the integration of the luminous intensity distribution over the solid angle subtended by the light source. In practice the flux is commonly calculated from the luminous intensity distribution of the light emitting component or luminaire.[2], [3] The luminous intensities are measured at various positions in space and its distribution tabulated in the form of discrete data.

Per the current industrial practice, the Zonal Lumen Method is used to calculate the total luminous flux [2,4]. This method computes for the flux without any interpolation for the given discrete data. The intensity is measured at the mid-zone to represent the zonal average, and the total flux is the product of intensities with the respective zonal constants. The zonal constant is a parameter that relates to the solid angle of the light radiated in the zone. (See Table 1)

Table 1

Zonal Lumen Method For Flux Computation: An Example

Zone Limit (degrees)	Zonal Constant, $C_z(x)$	Mid-point Intensity, $I_v(x)$	Zonal Flux = $I_v(x)\ .\ C_z(x)$
0-10	0.0955	1.7103	0.1633
10-20	0.2835	1.3455	0.3814
20-30	0.4629	0.9566	0.4428
30-40	0.6282	0.7316	0.4596
40-50	0.7744	0.7330	0.5676
50-60	0.8972	0.5435	0.4876
60-70	0.9926	0.3383	0.3358
70-80	1.0579	0.2857	0.3023
80-90	1.0911	0.2304	0.2514
		Total Flux	3.3916

Tables of Zonal Constants are listed in existing lighting handbooks and literatures for the convenience of computation for the end user, with various step sizes for the angular zones catering to light emission for beams with narrow angle to beams with wide angle.[2,5] As more measurement points are taken, the accuracy of the total computed flux increases but this comes at a cost. On the other hand, if too few measurements are taken, it may fail to capture the true radiation pattern of the light source.

For the purpose of this paper, we explore luminaires having symmetrical luminous intensity distribution about the beam axis, which forms a large class of light sources. Examples of such distributions are illustrated in Figure 3a, 4a and 5a. The luminous intensity distribution is typically plotted in Cartesian or Polar coordinates, and is derived from intensity measurement data at discrete angular positions with respect to the beam axis. Typical number of the light beam characterization measurements varies from tens to about a hundred data points, depending on the complexity of the beam pattern.

2. Luminous Flux Computation: Basis of Zonal Lumen Method

For a light source with intensity distribution symmetrical about the beam axis, the luminous flux equation, i.e. Equation (1), reduces to one variable, in θ, the angular position in which the luminous intensity is measured with respect the axis of the light beam, and is given by

$$\phi = \int_{\Omega} I_v d\Omega = 2\pi \int_0^{\pi/2} I_v(\theta).\sin\theta.d\theta \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

where θ is the angle measured from the axis of symmetry of the beam, and
$I_v(\theta)$ is the luminous intensity distribution function, represented by a table of values.

Equation (2) in its discrete form becomes

$$\text{Total Flux} = \sum_{m=1}^{n} I_v(\theta_m).(2\pi\sin\theta_m\Delta\theta) = \sum_{m=1}^{n} I_v(\theta_m).C_m(\theta) \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3)}$$

where $\Delta\theta$ is the angular step-size, equals to $(\frac{\pi}{2n})$
n is the number of steps
$C_m(\theta)$ is the Zonal Constant

Equation (3) in its discrete form is the basis for the Zonal Lumen Method, as described in photometry and illumination engineering literatures. [2,4]. The zonal constant for a luminaire with axis of symmetry about the beam axis, is given by:

$$C_m(\theta) = 2\pi[\cos\theta_m - \cos\theta_{m+1}] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4)}$$

where m is the step sequence, and
θ_m is the angle of measurement (with respect to the beam's axis of symmetry)

3. Optimal Method For Flux Computation – A Discussion

The flux integral, $2\pi \int_0^{\pi/2} I_v(\theta).\sin\theta.d\theta$, is an unknown function. Per current lighting industry practice, the intensity distribution function, $I_v(\theta)$, is determined approximately by sampling the beam intensity at discrete steps of angular position, θ, with respect to the light beam's axis. The angular positions are chosen, for the convenience of Zonal Lumen Method of computation, to match the published Zonal Constants tables. The positions of measurement are typically evenly spaced, starting 0° to 90° for symmetrical sources. The step size ranges from 0.1° for narrow angle beams to 10° for very wide angle beams.

The position of measurements would not have to be carried out at evenly spaced points, if one is not constrained by the Method of Zonal Lumen and the use of the Zonal Constant tables. The user will then have the flexibility of sampling more measurements near positions where the beam intensity is known to be highly irregular.

Our objective is to arrive at a method that is preferably be able to handle both evenly and unevenly spaced data. The spacing of the data refers to the angular position so chosen, θ.

For our consideration, we can broadly divide the methods available at our disposal to evaluate the flux integral as follows:

a) Numerical Integration
b) Analytic Integration
c) Hybrid Numeric-Analytic Method

The flux integrand, for convenience, is now restated as $h(x) = f(x).\sin x$, where $f(x)$ is unknown and is represented by sampled discrete data which is deemed to be exact. For this reason, it is preferable to interpolate the discrete data rather than using least squares fitting methods, for which the curve do not necessarily pass through the discrete data points.

a) Numerical Integration.

The classical numerical integration, also called quadratures, employ the strategy of approximating the curve $y = f(x)$ with a simpler function for easy integration.

Newton Cotes rules are obtained by approximating the function $f(x)$ with polynomial equations of degree n-1 to interpolate for n equally spaced data points i.e. the Trapezoidal Rule using linear function, Simpson's Rule using quadratics, Simpson's 3/8 rule using cubic polynomials and so on. The Composite Rule employs the strategy of dividing the unknown function into N segments, and applying the Newton-Cote rules to each segment and summing it up.

As the flux integrand is composed of a product of two functions i.e. an unknown $f(x)$ and a trigonometric term, $\sin x$, it is not easily amenable to computation using the Newton Cotes rule. Further, the requirement for evenly spaced data by the Newton Cotes rule put a constraint on flux computation for given experimental data that is not evenly spaced.

Gaussian quadratures are normally used for analytic functions but it has two features that make it unsuitable for evaluating the flux integral. Firstly, the flux integral is composed of discrete data but the Gaussian quadrature requires that the underlying function be known in order to evaluate the function at particular abscissas. This means that one have to interpolate between the points using an approximation analytic function, and use that function for piecewise composite Gaussian quadrature computation. Secondly it requires pre-selected abscissas and weights for approximation of the function under consideration. However, for flux measurements we have given discrete data that do not necessarily coincide with the abscissas required. As such, Gaussian Quadratures are not considered for evaluating the flux integral.

b) Analytic Integration.

Analytic integration is not directly applicable to the flux integrand until that integrand is reduced to an analytic expression. This is possible by reconstructing piecewise analytic functions, as the unknown function, $f(x)$ in the flux integrand $f(x).\sin x$ is represented by the discrete intensity data and can be approximated using polynomial interpolating functions. The task now is to select the most fitting interpolating function to approximate as closely as possible the underlying function of the discrete data set.

Some of the characteristics of the curve of the unknown function *f(x),* representing the luminous intensity distribution of light sources, that is desirable to be reproduced in the end analytic function are:

i) It is continuous.
ii) It has continuous first and second derivatives with no singularities.
iii) It may have many maxima and minima in it.
iv) Its shape may be mainly concave, or mainly convex, or a mixture of both.
v) It may have small local perturbations.

Polynomial Interpolation of *f(x).*

For interpolation of discrete data, we have the following interpolation schemes available for consideration:

i) Single polynomial, derived from either Lagrange Basis or Newton Basis.
ii) Piecewise linear interpolation.
iii) Piecewise quadratic interpolation.
iv) Piecewise Cubic Hermite polynomial
v) Piecewise Cubic Spline polynomial.

For single polynomial interpolation function, there is only one unique polynomial of degree n-1, passing through n nodes. Thus the polynomial derived from the Lagrange basis or from the Newton basis are of necessity the one and same polynomial despite its differing origin of derivation. Proof of this can be found in Conte and De Boor.[6]

The typical number of data points for determining the shape of the intensity distribution run into tens to about a hundred. As such, a single polynomial interpolation will require such a high order polynomial that it is unsuitable. It is a well-known phenomenon that high order polynomials tend to be highly unstable and can generate wild oscillation between data points. As such, single polynomial is not suitable for interpolating the discrete intensity data.[7]

Piecewise linear interpolation truncates the actual underlying function, and depending on the convexity and the number of extrema present in the function, the truncation error of linear interpolation can fluctuate wildly, sometimes self-compensating at other times accumulative. An example is a function that is a single concave or single convex shape. It fails to approximate curvatures well. Further, it suffers from lack of continuity in the first derivative at the nodes or knots and is visibly unnatural looking.

Mathematica's *Interpolation* function applied to discrete data set produces an InterpolationFunction object based on divided differences to construct Lagrange or Hermite interpolating polynomials. However, use of Hermite interpolating polynomial presumes a knowledge of the first derivative of $h(x) = f(x).sinx,$ otherwise some further means of approximation is needed to estimate the first derivative at the knot from the set of discrete data. Thus the Hermite interpolating function is not suitable for interpolating discrete data for flux computation.

Piecewise quadratic again lacks continuity in the first derivative at the knots and causes the curve to deviate from the true underlying function, and is not much better than its linear counterpart.

Piecewise cubic polynomials are much more useful, and in this regard the piecewise cubic Hermite polynomial have the property of agreeing in the function and its first derivative at each of the knots. This enable the piecewise cubic Hermite polynomials to be coupled together with a smooth joint, unlike the standard cubic polynomial, quadratic or the linear interpolating functions. But as said earlier, Hermite cubic polynomial requires prior knowledge of the first derivative at the knots.

Piecewise cubic spline polynomials extend the requirement to having the second derivative at each of the knot to agree with the two adjacent cubics. This provides another degree of "smoothness" to the joint. Thus for cubic spline, $f'_i(x_{i+1})=f'_{i+1}(x_{i+1})$, and $f''_i(x_{i+1})=f''_{i+1}(x_{i+1})$. These conditions, plus the natural end spline conditions where the second derivatives equals to zero, and enforcing the continuity at the joints for the function of the curve gives the necessary 4(n-1) simultaneous equations to solve for 4(n-1) unknown coefficients of the n-1 spline segments, for n given nodes.[7,8,9]

The cubic spline polynomial interpolation, by virtue of its requirement for continuity at the knots to agree up to the second derivative, is able to provide the smoothest reproduction of the curve that represents the underlying function of the luminous intensity data set, over and against the other interpolation schemes.

c) Hybrid Numeric-Analytic Method.

We have seen that the piecewise cubic spline polynomial can reconstruct and approximate closely the underlying function for the flux integrand, so by carrying out analytic integration of the piecewise analytic function and summing up the values of the piecewise integrals we would arrive at the total flux. This composite method of summation of the pieces is numerical in nature, while the integration portion is analytic in nature. The analytic interpolating function is an approximation procedure, and so we end up using a hybrid of numerical and analytic techniques to compute the total flux in order to have an optimal method of computing the total flux. (See Figure 1 and Figure 2.) As the given data set is discrete, we can never know the true value of the integral and the closest we can get to is to approximate as closely as possible the curve forming the flux's integrand.

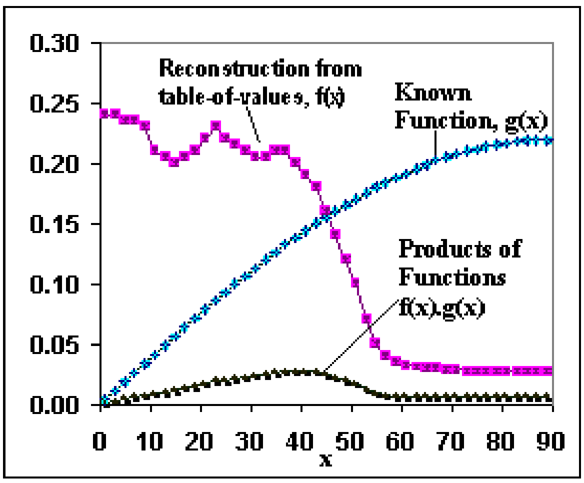

Figure 1 Product of Functions

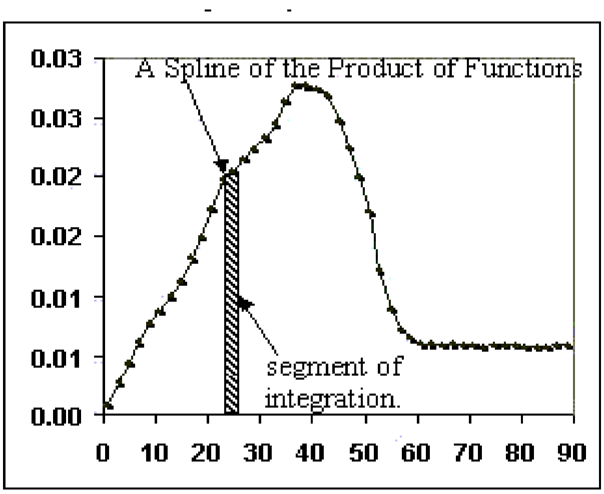

Figure 2 Piece-wise Analytic Integration

4. Piece-wise Integration of the Luminous Flux Integral

With the luminous intensity function reconstructed from its discrete data set, the intensity function is now represented by piece-wise polynomials i.e.

$$I_{v,i}(\theta) = c_{i,1} + c_{i,2}\theta + c_{i,3}\theta^2 + \ldots + c_{i,n}\theta^{n-1} = \sum_{j=1}^{n} c_{i,j}\theta^{j-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots(5)$$

where i is the i*th* piecewise polynomial of the intensity function with *m* segments.
and $C_{i,j}$, *j=1, 2, 3, …, n* are the coefficients of the i*th* segment's polynomial.

Substituting Equation (5) into Equation (2), we have Luminous Flux , ϕ_i , at the i*th* segment between the interval $\theta = \theta_{i-1}$ and $\theta = \theta_i$,

$$\phi_i = 2\pi \int_{\theta_{i-1}}^{\theta_i} I_v(\theta).\sin\theta.d\theta = 2\pi \int_{\theta_{i-1}}^{\theta_i} \sum_{j=1}^{n} c_{i,j}\theta^{j-1}\sin\theta.d\theta$$

Hence, summing up for the Total Luminous Flux, ϕ_{TOTAL} , for all segments,

$$\phi_{TOTAL} = 2\pi \sum_{i=1}^{m} [\int_{\theta_{i-1}}^{\theta_i} \sum_{j=1}^{n} (c_{i,j}\theta^{j-1}\sin\theta)d\theta]$$

Thus the Total Luminous Flux can be computed in a hybrid of numerical and analytic integration. This can be implemented in a mathematical software environment readily. Weighing the trade-off between computational efficiency and the oscillatory nature of higher order polynomials, we choose as a compromise *n=4* i.e. a cubic polynomial for the spline.[7] We then have the particular analytic solution for the Total Luminous Flux as

$$\phi_{TOTAL} = \sum_{i=1}^{m} \phi_i = 2\pi \sum_{i=1}^{m} [\int_{\theta_{i-1}}^{\theta_i} (c_{i,1}\sin\theta + c_{i,2}\theta\sin\theta + c_{i,3}\theta^2\sin\theta + c_{i,4}\theta^3\sin\theta)d\theta]$$

$$= 2\pi \sum_{i=1}^{m} \left[c_{i,1}[-\cos\theta] + c_{i,2}[\sin\theta - \theta\cos\theta] + c_{i,3}[2\theta\sin\theta + (2-\theta^2)\cos\theta] + c_{i,4}[(3\theta^2 - 6)\sin\theta + (6\theta - \theta^3)\cos\theta]\right]_{\theta_{i-1}}^{\theta_i}$$

In MATLAB, the coefficients for the splines segments are returned simply by calling the *unmkpp* function, e.g.

```
x= [0 1 2 3 4 5 6 7 8 9 10]       ←Data input
y=[ 0 4 3 5 8 6 5 4 3 1 2 0]
pp=spline (x,[0 y 0])              ← generates the cubic spline polynomials, 0 end-slopes specified.
[breaks,coefs, m,n,d]=unmkpp(pp)   ←extracts and prints information related to the cubic spline
                                     polynomial, including the coefficients.
```

Letting the unknown function $f_{12}(x)$ be the cubic spline polynomial between knot 1 and 2,
$f_{12}(x) = a + bx + cx^2 + dx^3$, where *a, b, c,* and *d* represents the coefficients' numerical values

Then

$h_{12}(x) = f_{12}(x).\ \sin x$

$= (a + bx + cx^2 + dx^3).\sin x$, *where* h_{12} *represent spline segment between knot 1 and 2.*

The expression, $(a + bx + cx^2 + dx^3).\sin x$, can be symbolically integrated and evaluated in MATLAB's Symbolic Math Toolbox
i.e.

```
Syms x      ←Declares symbolic variable x
Flux = int ((a + bx + cx^2 + dx^3).sin x, limit1, limit2)
```

and giving the solution in its analytic form, or value if the limits are supplied::

Flux = $[(a_1 + a_2.x + a_3.x^2).\sin x + (b_1 + b_2.x + b_3x^2 + b_4x^3.)\cos x]$, where a_i, b_i are constants.

Summing over all the segments gives us the total flux.

5. Numerical Experiments and Results

The Hybrid Numeric-Analytic Method is tested on several Luminous Intensity Distribution Functions (LIDF), and three LIDFs are reproduced here, represented by LIDF A, B and C (see Fig 3a, 4a and 5a, respectively).

LIDF A and B are actual measurements of some luminaires, while LIDF C is simulated using a known function that closely resembles a typical luminaire with smooth LIDF i.e. Runge's function[8]. LIDF A, B and C represent typical luminous intensity distributions where the intensities peaked at the beam's axis of symmetry, and reduces in intensity as the angle of the detector to the beam axis is increased, with the intensity dropping down to close to zero at a position perpendicular to the beam's axis. They also represent the typical distribution of luminaires, some with a monotonically decreasing function, represented by LIDF C, and some with localized irregularities represented by LIDF A and B.

The table of values for the LIDFs of these lamps has a total of 90 intensity values measured at an interval of 1 degree over one side of the symmetrical distribution about the lamp beam's axis. The test is carried out over sets of data with 18, 30, 45 and 90 measurements evenly divided over the 90 degrees range of measurement, and the total flux measurement is compared for the Numeric-Analytic Method against the Zonal Lumen Method.

As the luminous intensity distributions are represented by a finite number of intensity measurements, the value of the Total Luminous Flux will never be known exactly. However the nature of most luminaires, by virtue of its design and construction, is such that their LIDFs' irregular features are captured with about a hundred measurements for each plane of measurement. For beams with known rotational symmetry, only one plane of measurement is necessary. As such, the Total Luminous Flux computed using Zonal Lumen Method or Numeric-Analytic Method will tend to converge to close to exact value when the number of measurements used is about 100 or

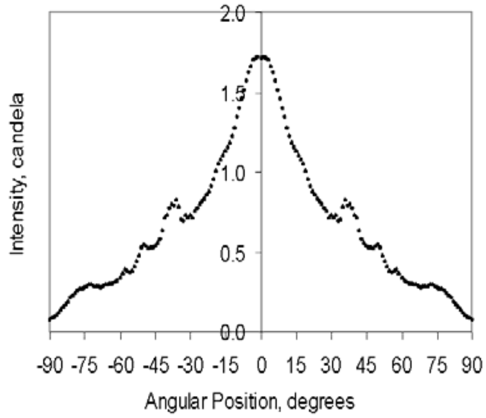

Fig 3a LIDF A

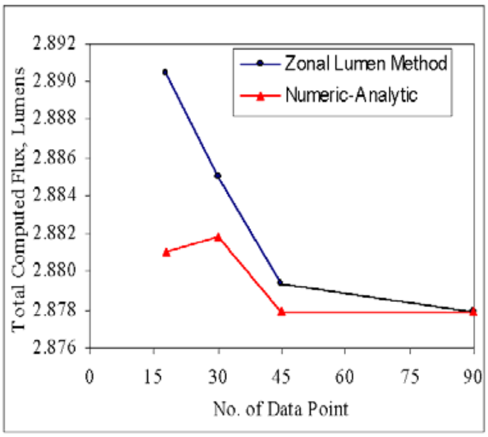

Fig 3b LIDF A - Computed Flux Vs No. of Data Points

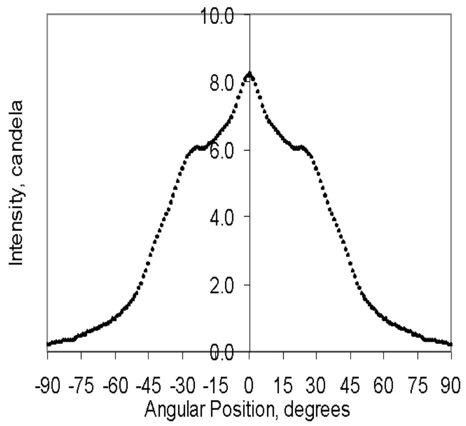

Fig 4a LIDF B

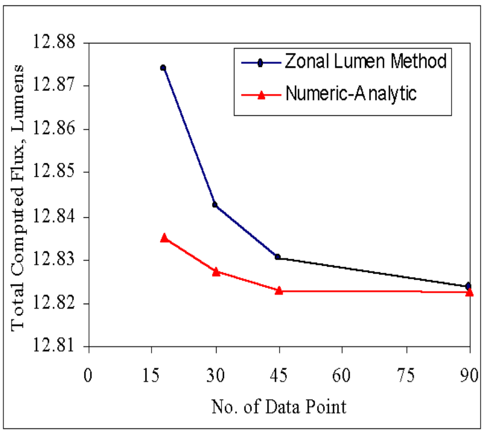

Fig 4b LIDF B - Computed Flux Vs No. of Data Points

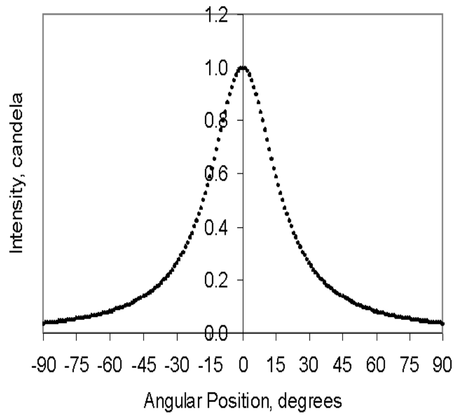

Fig 5a LIDF C

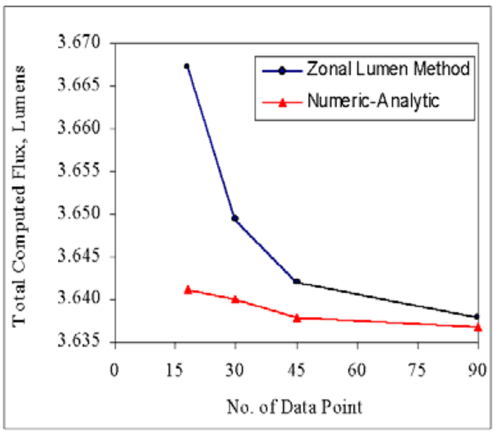

Fig 5b LIDF C - Computed Flux Vs No. of Data Points

more.The goal of this alternative computation using the Hybrid Numeric-Analytic Method is to provide a more economical means to achieve similar accuracies as the Zonal Lumen Method but using fewer data points.

The results of the numerical experiments are shown in Fig 3b, 4b and 5b, comparing the Zonal Lumen Method applied to the LIDF and the Trigonometric Function of Zonal Constant) with the Hybrid Numeric-Analytic Method. In general the Hybrid Numeric-Analytic Method converges to the exact Total Luminous Flux faster than the Zonal Lumen Method.

6. Conclusion

The Hybrid Numeric-Analytic Method in general is more economical in needing fewer measurement points as compared to the Zonal Lumen Method to approximate the exact Total Luminous Flux. It demonstrates to us that a combination of hybrid numerical and analytic methods (i.e. recovery of analytic functions from a table of values by piecewise cubic spline interpolation, and integration of analytic functions by symbolic mathematics) through the use of current mathematical software can be a powerful tool to enhance considerably the purely numerical methods. This is illustrated in the case of flux computation in the lighting industry and likely in other fields as well.

References

[1] McCluney, William R., 1994, *Introduction to Radiometry and Photometry*, Artech House

[2] Rea, Mark S., 2000, *The IESNA Lighting Handbook, Ninth Edition*, IESNA.

[3] *The Measurement of Luminous Flux*, CIE Publication No. 84, 1989.

[4] DeCusatis, Casimer, Ed., 1998. *Handbook of Applied Photome*try, AIP Press

[5] *Journal of Illuminating Engineering Society, 1972, 1(2):181-184. IES Committee on Testing Procedure. Subcommittee on Photometry of Indoor Luminaires*

[6] S.D. Conte and C, de Boor, 1980, *Elementary Numerical Analysis: An Algorithmic Approach,* McGraw-Hill, New York, 3rd edition.

[7] Press, William H, et al, 1992, *Numerical Recipes in C – The art of Scientific Computing*, Cambridge University Press.

[8] De Boor, Carl,1978, *A Practical Guide to Splines*, Springer-Verlag

[9] Heath, Michael T., 1997, *Scientific Computing : An Introductory Survey*, McGraw-Hill.