

On the optimum value of τ for a variant of the Diffusion Method: A Generalization

Gregory Karagiorgos^a, Giorgos Kollias^{b,1}, Nikolaos Missirlis^a,
Elias Tsigaridas^a

^a*Department of Informatics and Telecommunications, University of Athens
Panepistimioupolis, 15784 Athens, GREECE*

^b*Department of Computer Engineering and Informatics, University of Patras
26500 Patras, GREECE*

Abstract

In this paper we find an exact formula for the eigenvalues of the iteration matrix of the Generalized Diffusion method (GDF) used in load balancing. This provides us with the optimum value of the extrapolation parameter, i.e. the parameter endowing GDF with extra flexibility compared to other iterative methods. We apply our technique to the broad class of n-D torus interconnection topologies and specialize our main result in the case of rings, 2-D tori, n-D hypercubes and n-D, k-ary hypercubes. Simulation results are then presented demonstrating the validity of our theoretical approach.

Key words: Diffusion methods, load balancing, iterative methods, distributed computing, Fourier analysis, network graphs

1 Introduction

One of the most fundamental problems to solve on a parallel or distributed processor system is to balance the workload among its computational nodes in order to maximize efficiency. In this paper we focus on the Generalized Diffusion method (GDF) (Karagiorgos [6]), which is an iterative, distributed scheme for tackling the same problem.

Various diffusion methods are explored in literature. Cybenko [2] and Boilat [1] suggest experimentally justified optimal parameter values for the classical diffusion method. Xu and Lau ([10],[9]) find optimal parameter values for a range of topologies in the one flow parameter per processor case.

Our approach is actually a tool for computing optimum diffusion parameters for GDF by calculating a closed formula for the spectrum of the global iteration matrix. We closely follow the analysis of [7] providing a generalization of theoretical results presented therein. Our eigenvalue-generating method has the following characteristics:

General: it can be suitably applied to any processor network with nodes of identical environment both in terms of the number of links (degree) and corresponding flow parameters. So we can consider 2 and n different flow parameters for the 2-D torus and n -D hypercube respectively.

Simple: no special theoretical knowledge is necessary i.e. our approach does

Email addresses: `greg@di.uoa.gr` (Gregory Karagiorgos),
`gkollias@di.uoa.gr` (Giorgos Kollias), `nmis@di.uoa.gr` (Nikolaos Missirlis),
`et@di.uoa.gr` (Elias Tsigaridas).

¹ part of the work was prepared while a postgraduate student at the Department of Informatics and Telecommunications, University of Athens

not use circulant matrix theory to compute the eigenvalues of the iteration matrix (Davis [3]).

It stresses GDF's main virtue: given a topology with fixed link characteristics (i.e. c_{ij}) the τ parameter of GDF is the only option of accelerating a possibly slow (and unchangeable) flow parameter set. So it is important to have a method to provide the best possible value of τ .

The rest of the paper is organized as follows: In Section 2 we give a general overview of GDF. In Section 3 we introduce the key notion behind the technicalities to follow. In Section 4 we apply the idea to the n-D torus topology class: the key notion is further formalized culminating in a proof of our main theoretical result. Section 5 is a brief account of applying our result to standard n-D torus interconnection topologies. Section 6 introduces simulation graphs validating our closed-formula-type results of sections 4,5. Finally, Section 7 concludes the paper briefing advantages of our method.

2 The Generalized Diffusion Method (GDF)

GDF is a synchronous, homogeneous load balancing scheme given by

$$u_i^{(n+1)} = u_i^{(n)} - \tau \sum_{j \in A(i)} c_{ij} (u_i^{(n)} - u_j^{(n)}) \quad (1)$$

where u_i is the load of node i , c_{ij} 's are the flow parameters e.g link capacities, $A(i)$ is the set of the nearest neighbors of node i and superscripts $(n, n+1)$ denote iteration steps and τ is the accelerating parameter.

In matrix notation GDF can be written as

$$u^{(n+1)} = Mu^{(n)}, \quad (2)$$

where $M = (m_{ij})$ is the generalized diffusion matrix. Alternatively

$$u^{(n+1)} = (I - \tau L)u^{(n)}, \quad (3)$$

where L is known as the weighted Laplacian matrix of our graph. Conditions for convergence of GDF are well known (Karagiorgos [6]).

3 Spectrum and Optimal Convergence Rate

3.1 Notation

We will use Dirac notation (Dennerly et al. [4]). Here $|a\rangle$ denotes a vector, $\mathfrak{B} = \{|j\rangle, j = 1, 2, \dots, n\}$ a basis, $\langle b|a\rangle$ the inner product and $\hat{T}$ an operator. The identity operator can be written as $\hat{I} = \sum_{j=1}^n |j\rangle\langle j|$ and matrix representation of $\hat{T}$ then is

$$(T)_{ij} = \langle i|\hat{T}|j\rangle, \quad i, j = 1, 2, \dots, n. \quad (4)$$

So GDF can be formulated as follows

$$|u\rangle^{(n+1)} = \hat{M} |u\rangle^{(n)}. \quad (5)$$

We investigate the representation of $\hat{L}$ operator for two different basis selections for the n-D torus family of topologies, compute its spectrum and specialize our results for some example cases, namely

- ring
- 2-D torus
- n-D hypercube
- n-D, k-ary hypercube

In the first basis, $\hat{L}$ will be represented by a matrix readily generated by the topology-respective M matrix. But there is no direct closed form for computing its eigenvalues. However, the other suitably selected basis will yield a diagonal-matrix representation of $\hat{L}$ so the spectrum of this similar matrix is easily attained. Then, the optimum value τ_{opt} of the extrapolation parameter of GDF can be computed by (Varga [8])

$$\tau_{opt} = \frac{2}{\lambda_2 + \lambda_n} \quad (6)$$

where λ_2, λ_n are respectively the second smallest and the maximum eigenvalue of $\hat{L}$ operator.

4 n-D torus

We consider the n-D torus consisting of N_m , $m = 1, \dots, n$ processors per dimension, giving a total of $p = \prod_{m=1}^n N_m$ processors. The position of each processor within the torus is encoded through its $i_1 i_2 \dots i_n$ -label, $i_m = 1, \dots, N_m$. Thus $u_{i_1 i_2 \dots i_n} = \langle x_{i_1 i_2 \dots i_n} | u \rangle$ is $i_1 i_2 \dots i_n$ -labelled processor's load. In other words it is the projection of the load distribution $|u\rangle$ against the positional vector $|x_{i_1 i_2 \dots i_n}\rangle$. It follows, that the set of all $u_{i_1 i_2 \dots i_n}$ constitutes the components of $|u\rangle$ under basis $\mathfrak{B}_1 = \{|x_{i_1 i_2 \dots i_n}\rangle, i_m = 1, \dots, N_m\}$.

The Generalized Diffusion Method (GFD) can be described as a series of steps

$$|u\rangle^{(n+1)} = \hat{M}_\tau |u\rangle^{(n)}, \quad n = 0, 1, \dots \quad (7)$$

At each step, a new load distribution $|u\rangle^{(n+1)}$ is generated out of an old one $|u\rangle^{(n)}$. $\hat{M}_\tau = \hat{I} - \tau \hat{L}$ is known as the GDF iterative operator and τ is a global parameter, which when suitably selected in value accelerates our method. $\hat{I}$

and $\hat{L}$ are the identity and the Laplacian operator respectively. In the sequel we will look into the presentation of $\hat{L}$ under two different bases. One of them will be the naturally occurring $\mathfrak{B}_1$, whilst the other will comprise of vectors in

$$\mathfrak{B}_2 = \{|k_{j_1 j_2 \dots j_n}\rangle, j_m = 0, 1, \dots, N_m\} \quad (8)$$

Transition between these two basis-induced views of our mathematical objects is effectively carried out through use of the following p^2 inner products

$$\langle x_{i_1 i_2 \dots i_n} | k_{j_1 j_2 \dots j_n} \rangle = \frac{1}{\sqrt{p}} e^{i \sum_{m=1}^n \frac{2\pi}{N_m} (j_m - 1)(i_m - 1)}, \quad i_m, j_m = 1, \dots, N_m, \quad m = 1, \dots, n \quad (9)$$

In the case of $\hat{L}$ operator, these two views are highly complementary; applying $\hat{L}$ on vectors of $\mathfrak{B}_1$ naturally reduces to a direct expression of the communication pattern between a processor and its immediate neighbors. On the other hand, representation of $\hat{L}$ in $\mathfrak{B}_2$ is proved to be a diagonal matrix. In other words, while $\hat{L}$ is easily understood under $\mathfrak{B}_1$, it readily reveals its spectrum only under $\mathfrak{B}_2$. In what follows we will produce a closed formula giving the spectrum of $\hat{L}$. This permits us to pinpoint the optimum value for the acceleration parameter τ , i.e. the parameter value leading to the fastest possible convergence of GDF under arbitrary initial load distributions.

From this point on, we will employ the notations

$$\begin{aligned}
\underline{i} &= i_1 i_2 \dots i_n \\
\underline{j} &= j_1 j_2 \dots j_n \\
\Sigma_{\underline{i}} &= \Sigma_{i_1=1}^{N_1} \Sigma_{i_2=1}^{N_2} \dots \Sigma_{i_n=1}^{N_n} \\
\underline{i}_{m\oplus 1} &= i_1 i_2 \dots i_{m\oplus 1} \dots i_n \\
\underline{i}_{m\ominus 1} &= i_1 i_2 \dots i_{m\ominus 1} \dots i_n
\end{aligned} \tag{10}$$

where $\oplus, \ominus$ denote suitable modulo additions and subtractions. It is straightforward that

$$\hat{L}|x_{\underline{i}}\rangle = 2 \sum_{m=1}^n f_m c_m |x_{\underline{i}}\rangle - \sum_{m=1}^n f_m c_m (|x_{\underline{i}_{m\oplus 1}}\rangle + |x_{\underline{i}_{m\ominus 1}}\rangle) \tag{11}$$

where the first term represents exported load through the pair of immediate neighbors along each of n dimensions. The second term is simply imported load from each of these two neighbors, $c_m, m = 1, \dots, n$ are flow parameters regulating communication bandwidth on a per-dimension basis, while

$$f_m = \begin{cases} 1, & \text{if } N_m > 2 \\ \frac{1}{2}, & \text{if } N_m = 2 \end{cases}. \tag{12}$$

The parameter takes into consideration the case of two processors along at least one dimension, where wrapping around would introduce a double connection between them, and this is not allowed in our simple graph interconnection model.

We can now prove the following Theorem.

Theorem 1 *The eigenvalues of $\hat{L}$ are given by*

$$\lambda_{i_1 i_2 \dots i_n} = \lambda_{\underline{i}} = 4 \sum_{m=1}^n f_m c_m \sin^2 \frac{\pi}{N_m} (i_m - 1). \quad (13)$$

Proof Applying $\hat{L}$ to basis $\mathfrak{B}_2$ vectors we obtain:

$$\begin{aligned} \hat{L} |k_{j_1 j_2 \dots j_n}\rangle &= \hat{L} |k_{\underline{j}}\rangle = \hat{L} \left(\sum_{\underline{i}} |x_{\underline{i}}\rangle \langle x_{\underline{i}}| \right) |k_{\underline{j}}\rangle = \\ &= \sum_{\underline{i}} \langle x_{\underline{i}} | k_{\underline{j}} \rangle \hat{L} |x_{\underline{i}}\rangle \stackrel{(11)}{=} \sum_{m=1}^n A_m (C - R_m - L_m) \end{aligned} \quad (14)$$

where $A_m = 2f_m c_m$, $C = \sum_{\underline{i}} \langle x_{\underline{i}} | k_{\underline{j}} \rangle |x_{\underline{i}}\rangle$, $m = 1, \dots, n$ and $R_m = \sum_{\underline{i}} \langle x_{\underline{i}} | k_{\underline{j}} \rangle |x_{\underline{i}_{m\oplus 1}}\rangle$ and $L_m = \sum_{\underline{i}} \langle x_{\underline{i}} | k_{\underline{j}} \rangle |x_{\underline{i}_{m\ominus 1}}\rangle$ are the $2n$ terms stemming, respectively from “right”- and “left”-neighbor communication along each dimension. First we’ll compute R_m for arbitrary m :

$$\begin{aligned} R_m &= \sum_{\underline{i}} \langle x_{\underline{i}} | k_{\underline{j}} \rangle |x_{\underline{i}_{m\oplus 1}}\rangle = \sum_{\underline{r}} \sum_{\underline{i}} \langle k_{\underline{r}} | x_{\underline{i}_{m\oplus 1}} \rangle \langle x_{\underline{i}} | k_{\underline{j}} \rangle |k_{\underline{r}}\rangle \stackrel{(9)}{=} \\ &\stackrel{(9)}{=} \sum_{\underline{r}} \sum_{\underline{i}} \frac{1}{\sqrt{p}} e^{-i \sum_{s=1}^n \frac{2\pi}{N_s} (r_s - 1)(i_s - 1)} \times e^{-i \frac{2\pi}{N_m} (r_m - 1)(i_m \oplus 1 - 1)} \times \\ &\times \frac{1}{\sqrt{p}} e^{i \sum_{s=1}^n \frac{2\pi}{N_s} (j_s - 1)(i_s - 1)} |k_{\underline{r}}\rangle = \\ &= \frac{1}{p} \sum_{\underline{r}} e^{-i \frac{2\pi}{N_m} (r_m - 1)} \sum_{\underline{i}} \prod_{s=1}^n e^{i \frac{2\pi}{N_s} (i_s - 1)(j_s - r_s)} |k_{\underline{r}}\rangle = \\ &= \frac{1}{p} \sum_{\underline{r}} e^{-i \frac{2\pi}{N_m} (r_m - 1)} \prod_{s=1}^n N_s \delta_{j_s r_s} |k_{\underline{r}}\rangle = \\ &= e^{-i \frac{2\pi}{N_m} (r_m - 1)} |k_{\underline{j}}\rangle \end{aligned}$$

Applying the same technique, i.e. suitably inserting the identity operator $\hat{I} = \sum_{\underline{r}} |k_{\underline{r}}\rangle \langle k_{\underline{r}}|$ within expressions for L_m and C we can analogously obtain:

$$C = |k_{\underline{j}}\rangle \quad \text{and} \quad L_m = e^{i \frac{2\pi}{N_m} (r_m - 1)} |k_{\underline{j}}\rangle$$

Using these results in (14) we obtain the eigenvalues of $\hat{L}$ as

$$\lambda_{\underline{i}} = 2 \sum_{m=1}^n f_m c_m - 2 \sum_{m=1}^n f_m c_m \cos \frac{2\pi}{N_m} (i_m - 1)$$

which reduces to the sought for expression (13) if we apply the trigonometric identity $1 - \cos \phi = 2 \sin^2 \frac{\phi}{2}$ $\square$

Formula (13) is a concise way of finding all eigenvalues of $\hat{L}$ operator. However, the truth is that only λ_2 and λ_{max} , i.e. the second smallest and the maximum eigenvalues are needed in order to compute the optimum value of the acceleration parameter of our GDF method through (6). By inspecting (13) it is easy to verify that

$$\lambda_2 = 4 \min_m \left\{ f_m c_m \sin^2 \frac{\pi}{N_m} \right\} \quad \text{and}$$

$$\lambda_{max} = 4 \sum_{m=1}^n f_m c_m \sin^2 \frac{\pi}{N_m} \left\lfloor \frac{N_m}{2} \right\rfloor$$

Consequently (6) gives

$$\tau_{opt} = \frac{1}{2} \frac{1}{\sum_{m=1}^n f_m c_m \sin^2 \frac{\pi}{N_m} \left\lfloor \frac{N_m}{2} \right\rfloor + \min_m \left\{ f_m c_m \sin^2 \frac{\pi}{N_m} \right\}} \quad (15)$$

5 Example topologies

We will now specialize the result in (15), for four classes of interconnection topologies, namely ring, 2-D torus, n-D hypercube and n-D, k-ary hypercube.

5.1 Ring

Here $n = 1$, $p = N_1 = N$, $f_1 = 1$ for $N > 2$ and $c_1 = c$. Formula (15) yields

$$\tau_{opt} = \begin{cases} \frac{1}{2c} \frac{1}{1 + \sin^2 \frac{\pi}{N}} & , \text{ if } N \text{ is even} \\ \frac{1}{2c} \frac{1}{\sin^2 (1 - \frac{1}{N}) \frac{\pi}{2} + \sin^2 \frac{\pi}{N}} & , \text{ if } N \text{ is odd} \end{cases}. \quad (16)$$

5.2 2-D torus

Here $n = 2$, $p = N_1 \times N_2$, $f_m = 1$ for $N_m > 2$, $m = 1, 2$ and we have two flow parameters c_1, c_2 describing relative load mobility along the horizontal and vertical dimension respectively. Then formula (15) simplifies e.g. in the case when there exists $m^* \in \{1, 2\} : N_{m^*} = \max \{N_1, N_2\}$ and $c_{m^*} = \min \{c_1, c_2\}$

$$\tau_{opt} = \frac{1}{2} \frac{1}{c_1 \sin^2 \frac{\pi}{N_1} \lfloor \frac{N_1}{2} \rfloor + c_2 \sin^2 \frac{\pi}{N_2} \lfloor \frac{N_2}{2} \rfloor + c_{m^*} \sin^2 \frac{\pi}{N_{m^*}}}. \quad (17)$$

As an example of matrix representation of $\hat{L}$ under $\mathfrak{B}_1$ for a 4×4 torus, see Figure 1, where $\times = 2(c_1 + c_2)$, $\circ = -c_1$, $\bullet = -c_2$, and empty positions are null.

5.3 n-D hypercube

Here $p = 2^n$, $N_m = 2$, $f_m = \frac{1}{2}$, $m = 1, \dots, n$. Substituting in formula (15) we obtain

$$\tau_{opt} = \frac{1}{\sum_{m=1}^n c_m + \underline{c}_m}. \quad (18)$$

where $\underline{c}_m = \min_m \{c_m\}$. As an example of matrix representation of $\hat{L}$ under $\mathfrak{B}_1$ for a 3-D hypercube, see Figure 2, where $\times = c_1 + c_2 + c_3$, $\circ = -c_1$, $\bullet = -c_2$,

$\star = -c_3$ and empty positions are zeroes.

5.4 n -D, k -ary hypercube

Here $p = k^n$, $N_m = k$, $f_m = 1$, $m = 1, \dots, n$ for $k > 2$. We can distinguish the following two subcases:

- $k = 2s$, $s = 2, 3, \dots$. Substituting in formula (15) we get

$$\tau_{opt} = \frac{1}{2} \frac{1}{\sum_{m=1}^n c_m + \sin^2 \frac{\pi}{k} \underline{c}_m}. \quad (19)$$

- $k = 2s + 1$, $s = 1, 2, \dots$. Then

$$\tau_{opt} = \frac{1}{2} \frac{1}{\sum_{m=1}^n c_m \sin^2 \left(1 - \frac{1}{k}\right) \frac{\pi}{2} + \sin^2 \frac{\pi}{k} \underline{c}_m}. \quad (20)$$

6 Experimental Results

We used random initial load distributions for varying number of nodes and flow parameters and computed the number of iterations for GDF to converge in hypercubes. We remark that the theoretically optimum value of τ coincides exactly with the experimental ones. Similar results can be obtained for 2-D torus (see Figure 3).

7 Conclusions

We provide a method for the exact determination of the optimum value τ_{opt} of the accelerating parameter of GDF. This is accomplished through a technique

for computing the eigenvalues of the global iteration matrix of GDF. We can apply this method for any interconnection topology belonging to the n-D torus family. Our results generalize those e.g. of Xu and Lau, where it is assumed that $c_1 = c_2$ (2-D torus case); note that if $c_1 = c_2$ then (13) coincides with the one given in [9]. Moreover, for the approximate eigenvalues of M using local Fourier analysis (Karagiorgos [5]), it is proved that they are exact.

References

- [1] J. E. Boillat, Load balancing and Poisson equation in a graph, *Concurrency: Practice and Experience* **2**, 1990, 289-313.
- [2] G. Cybenko, Dynamic load balancing for distributed memory multi-processors, *J. Parallel Distrib. Comp.* **7**, 1989, 279-301.
- [3] P.J. Davis, *Circulant Matrices*, John Wiley and Sons, 1979
- [4] P. Dennerly and A. Krzywicki, *Mathematics for Physicists*, Dover Publications, 1996.
- [5] G. Karagiorgos and N. M. Missirlis, A local spectral approach for the load balancing problem, *Submitted for Publication*, 2004.
- [6] G. Karagiorgos, *Distributed diffusion algorithms for the load balancing problem*, PhD thesis, Athens, 2002 (in Greek)
- [7] Gregory Karagiorgos, George Kollias, Nikolaos Missirlis, Elias Tsigaridas, On the optimum value of τ for a variant of the Diffusion Method, *2nd MIT Conference on Computational Fluid and Solid Mechanics*, 2003.
- [8] R. Varga, *Matrix iterative analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

- [9] C. Z. Xu and F. C. M. Lau, *Load balancing in parallel computers: Theory and Practice*, Kluwer Academic Publishers, Dordrecht, 1997.
- [10] C. Z. Xu and F. C. M. Lau, Optimal parameters for load balancing with the diffusion method in mesh networks, *Parallel Processing Letters*, 4(2):139-147, 1994

Fig. 1. Laplacian Matrix for a 4×4 torus

Fig. 2. Laplacian Matrix for a 3-d hypercube

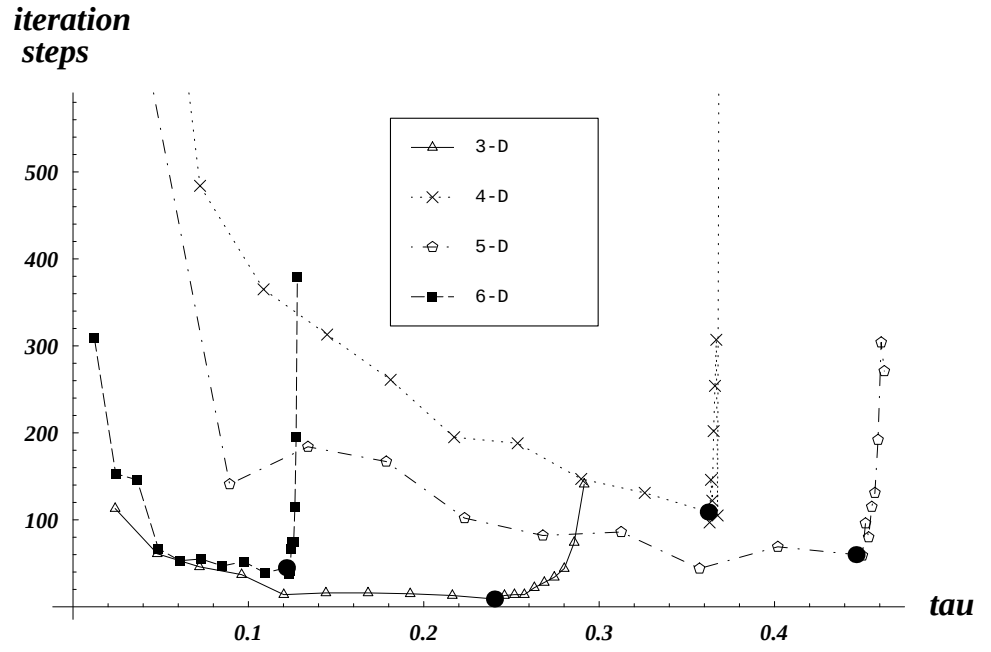


Fig. 3. **Hypercube:** Number of iterations for convergence versus τ . A filled circle denotes the theoretically optimum value τ_{opt} computed by Eq (6)