

**NASA Contractor Report 187458**

**ICASE Report No. 90-74**

# **ICASE**

## **QUADRATURE IMPOSITION OF COMPATABILITY CONDITIONS IN CHEBYSHEV METHODS**

**D. Gottlieb**

**C. L. Streett**

Contract Nos. NAS1-18107, NAS1-18605

October 1990

Institute for Computer Applications in Science and Engineering

NASA Langley Research Center

Hampton, Virginia 23665-5225

Operated by the Universities Space Research Association



National Aeronautics and  
Space Administration

Langley Research Center  
Hampton, Virginia 23665-5225

(NASA-CR-187458) QUADRATURE IMPOSITION OF  
COMPATIBILITY CONDITIONS IN CHEBYSHEV  
METHODS Final Report (ICASE) 21 p CSCL 20D

N91-10251

Unclass

G3/34 0310827



# QUADRATURE IMPOSITION OF COMPATIBILITY CONDITIONS IN CHEBYSHEV METHODS

D. Gottlieb<sup>1</sup>

Division of Applied Mathematics

Brown University

Providence, RI 02912

and

C. L. Streett

NASA Langley Research Center

Hampton, VA 23665

## ABSTRACT

Often, in solving an elliptic equation with Neumann boundary conditions, a compatibility condition has to be imposed for well-posedness. This condition involves integrals of the forcing function.

When pseudospectral Chebyshev methods are used to discretize the partial differential equation, these integrals have to be approximated by an appropriate quadrature formula. The Gauss-Chebyshev (or any variant of it, like the Gauss-Lobatto) formula can not be used here since the integrals under consideration do not include the weight function. A natural candidate to be used in approximating the integrals is the Clenshaw-Curtis formula, however we show in this paper that this is the wrong choice and it may lead to divergence if time dependent methods are used to march the solution to steady state.

We develop, in this paper, the correct quadrature formula for these problems. This formula takes into account the degree of the polynomials involved. We show that this formula leads to a well conditioned Chebyshev approximation to the differential equations and that the compatibility condition is automatically satisfied.

---

<sup>1</sup>Research was supported in part by the National Aeronautics and Space Administration under NASA Contract Nos. NAS1-18107 and NAS1-18605 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665. Research was also supported by the Air Force Office of Scientific Research grant no. AFOSR-90-0093, by DARPA-URI Contract N00014-86-K0754, and by NSF grant DMS-88-10150.



# INTRODUCTION

We deal here with a problem encountered in the solution via Chebyshev spectral collocation discretization of the Poisson equation with homogeneous Neumann boundary conditions. The problem arose in the context of solution of the pressure Poisson equation in a time-split algorithm for the incompressible Navier-Stokes equations. For this problem to be well-posed, the source term must satisfy a compatibility condition; the numerical analog of this condition using straightforward Clenshaw-Curtis quadrature formulae was found to be numerically ill-conditioned and produced large distortions in the spectral solution.

In Section I, we outline the time-split algorithm and describe the difficulty encountered; we proceed in Sections II and III to analyze the difficulty, first in terms of an equivalent parabolic equation which would be utilized, for instance, in the iterative solution of the Poisson equation of interest, then in terms of the steady equation itself. The proper quadrature formula is also developed, which alleviates the numerical difficulty. In Section IV we show numerical examples to demonstrate the numerical inconsistency and its resolution.

## I. TIME-SPLIT ALGORITHM

A splitting method is employed in many simulations to advance the solution of the incompressible Navier-Stokes equations from time  $t^n$  to  $t^{n+1}$ . Writing the Navier-Stokes equations in vector notation, with  $\mathbf{u}$  representing the velocity  $(u, v, w)$  we have:

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} \quad (1.1a)$$

and

$$\nabla \cdot \mathbf{u} = 0 \quad (1.1b)$$

in the domain  $\Omega$ , and :

$$\mathbf{u} = 0 \text{ on the boundary } \Gamma$$

The split scheme first advances  $\mathbf{u}^n$  to an intermediate solution  $\mathbf{u}^*$  by solving:

$$\mathbf{u}_t^* + \mathbf{u}^* \cdot \nabla \mathbf{u}^* = \nu \nabla^2 \mathbf{u}^* \quad (1.2)$$

$$\mathbf{u}^* = \mathbf{g}^* \text{ on } \Gamma$$

The intermediate boundary condition  $\mathbf{u}^* = \mathbf{g}^*$  is discussed in [1]. Finally, the solution is advanced from  $\mathbf{u}^*$  to  $\mathbf{u}^{n+1}$  via:

$$\begin{aligned} \mathbf{u}_t^{n+1} &= -\nabla P^{n+1} \\ \nabla \cdot \mathbf{u}^{n+1} &= 0 \end{aligned} \quad (1.3)$$

$$\hat{n} \cdot \mathbf{u}^{n+1} = 0 \text{ on } \Gamma$$

where  $\hat{n}$  is the unit normal to the boundary  $\Gamma$ . Note that the final, "pressure correction" step by itself is a set of inviscid equations; and is well-posed when boundary conditions on the normal component of the velocity only are enforced. At the end of the full step there exists a non zero tangential component of velocity on the boundary. The magnitude of this slip velocity can be reduced to  $O(\Delta t^3)$  by a proper choice of the intermediate boundary condition on  $\mathbf{u}^*$  [1].

Using backward Euler time discretization for Eq. (1.3) yields:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla P^{n+1} \quad (1.4)$$

The pressure step is actually carried out in two parts. First, the divergence of Eq. (1.4) yields:

$$\nabla^2 P^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (1.5)$$

where  $\nabla \cdot \mathbf{u}^{n+1} = 0$  is enforced. Then the velocities are updated using Eq. (1.4).

Note that this formulation requires a boundary condition for the pressure. This poses a problem, since there is no natural boundary condition for pressure. The use of a condition derived from enforcing the normal momentum equation at the boundary was attempted, but was apparently inconsistent as it resulted in explosive instability. Fortunately, analysis of the split scheme itself yields a self-consistent pressure condition:

$$\hat{n} \cdot \nabla P^{n+1} |_{\Gamma} = 0 \quad (1.6)$$

since both  $\hat{n} \cdot \mathbf{u}^*$  and  $\hat{n} \cdot \mathbf{u}^{n+1}$  are zero on the boundary. The error involved in this specification is, we believe, related to the overall splitting error of the scheme. It is also known that the error due to imposition of an inconsistent Neumann pressure boundary condition is isolated to a thin "boundary layer" [4].

It is easily shown using the divergence theorem that

$$\int_{\Omega} (\nabla \cdot \mathbf{u}^*) d\Omega = 0 \quad (1.7)$$

is required for the pressure Poisson equation to be well-posed with this boundary condition. For application of this algorithm in closed-system problems with homogeneous Dirichlet velocity boundary conditions, numerical tests indicate Eq. (1.7) is satisfied in general. However, in applications of a recently-developed non-reflecting outflow boundary treatment [5] to simulations of flow-through systems, it was found that Eq. (1.7) did not hold discretely for the intermediate velocity field. As will be shown in the numerical examples of Section IV, this leads to large distortions in the computed pressure field. It was decided that since

there is no solution to the problem:

$$\begin{aligned}\nabla^2 P &= \text{const} \neq 0 \quad \text{in } \Omega \\ \nabla P \cdot \hat{n} &= 0 \quad \text{on } \Gamma\end{aligned}$$

the pressure Poisson equation would be modified to read:

$$\nabla^2 P = \frac{1}{\Delta t} \left[ \nabla \cdot u^* - \int_{\Omega} (\nabla \cdot u^*) d\Omega \right] \quad (1.8)$$

The integral in Eq. (1.8) was implemented using the straightforward Clenshaw-Curtis quadrature formulae, appropriate for Chebyshev collocation. Although the pressure field distortions were much reduced, their magnitude was still unacceptable. The analysis presented in the following show the inconsistency in the use of these formulae, and the proper quadrature formula to recover a smooth solution to Eq. (1.8).

## II. TIME DEPENDENT PROBLEMS

Consider the parabolic equation

$$u_t = u_{xx} + g \quad (2.1a)$$

$$u(x, 0) = 0 \quad (2.1b)$$

with the Neumann boundary conditions

$$u_x(1, t) = u_x(-1, t) = 0. \quad (2.2)$$

We assume also that

$$\int_{-1}^1 g(x) dx = 0 \quad (2.3)$$

so that

$$\frac{d}{dt} \int_{-1}^1 u(x, t) dx = u_x(1) - u_x(-1) + \int_{-1}^1 g dx = 0 \quad (2.4)$$

and therefore in view of (2.1b)

$$\int_{-1}^1 u(x, t) dx = 0 \text{ for all } t. \quad (2.5)$$

In the pseudospectral Chebyshev method we seek an  $x$  polynomial  $u_N(x, t)$ , that satisfies the boundary conditions (2.2), such that (2.1a) is satisfied at the points  $x_j = \cos \frac{\pi j}{N}$ , namely we seek  $u_N(x, t)$  such that

$$\frac{\partial u_N}{\partial t} = \frac{\partial^2 u_N}{\partial x^2} + g \text{ at } x = x_j, j = 1, \dots, N-1 \quad (2.6a)$$

$$u_N(x, 0) = 0 \quad (2.6b)$$

and

$$\frac{\partial u_N}{\partial x}(-1, t) = \frac{\partial u_N}{\partial x}(1, t) = 0. \quad (2.7)$$

We refer the reader to [3] for a discussion of the stability and convergence of (2.6) and (2.7) to (2.1) and (2.2).

Here we are interested in the question whether the numerical approximation satisfies the conservation property (2.5). Since the numerical solution  $u_N(x, t)$  defined in (2.6) is a polynomial of degree  $N$  in  $x$  it is natural to consider the Clenshaw Curtis quadrature formula [2]. It uses the collocation points of (2.6a) and it is exact for polynomials of degree  $N$ .

**Lemma (1.1).** Let  $N$  be an even number and let  $\alpha_l$  be defined by

$$\alpha_e = \frac{4}{N} \sum_{k=0}^{\frac{N}{2}} \frac{1}{1-4k^2} \cos \frac{2\pi l k}{N} \cdot \frac{1}{c_k} \quad 0 < l < N \quad (2.8)$$



$$\alpha_0 = \alpha_N = \frac{1}{N^2 - 1} \quad \text{where } c_0 = c_N = 2 \quad c_k = 1 \quad 0 \neq k \neq N$$

Then for any polynomial  $h(x)$  of degree at most  $N$

$$\int_{-1}^1 h(x) dx = \sum_{l=0}^N h(x_l) \alpha_l \quad (2.9)$$

where  $x_l = \cos \frac{\pi l}{N}$ .

A close inspection of (2.6a) reveals that, in general,  $u_N(x, t)$  does not satisfy the conservation condition (2.5). Indeed, since  $g(x)$  in (2.1) is not a polynomial of degree  $N$ , the quadrature formula (2.9) is not exact for  $g$ , whereas it is exact for the other terms in the equation.

To remedy the situation it is customary to modify (2.6a) by seeking  $u_N(x, t)$  that satisfies

$$\frac{\partial u_N}{\partial t} + \frac{\partial^2 u_N}{\partial x^2} = g - \frac{1}{2} \sum_{l=0}^N g(x_l) \alpha_l \quad \text{at } x = x_j \quad 1 < j < N \quad (2.6c)$$

where  $\alpha_l$  are defined in (2.8). Of course  $u_N(x, t)$  still satisfies (2.1b) and (2.2). Thus the right hand side of (2.6c) has zero mean if the Clenshaw-Curtis formula is being used. Unfortunately, even with the above modification the solution of (2.6c) does not satisfy the discrete analog of (2.5); in fact we can state:

**Lemma (1.2):** Let  $u_N(x, t)$  be the solution of (2.6c,b) with the Neumann boundary condition (2.7). Let  $\hat{g}_N(t)$  be the coefficient of the  $N$ 'th Chebyshev polynomial in the expansion of  $g(x, t)$ , namely

$$\hat{g}_N = \frac{1}{\pi} \int_{-1}^1 \frac{g(x) T_N(x)}{\sqrt{1-x^2}} dx \quad (2.10)$$

$$T_N(x) = \cos N(\cos^{-1} x). \quad (2.11)$$

Then

$$\sum_{j=0}^N u_N(x_j, t) \alpha_j = \frac{2N}{N^2 - 1} \left\{ \hat{u}_N(t) - \int_0^t \hat{g}_N(\tau) d\tau \right\} \quad (2.12)$$

where

$$\hat{u}_N(t) = \frac{1}{\pi} \int_{-1}^1 \frac{u_N(x, t) T_N(x)}{\sqrt{1-x^2}} dx. \quad (2.13)$$

**Remark:** Equation (2.12) points out the difficulty in the method (2.6c). Suppose for example that  $g$  is a time independent function, then the solution of (2.1a,b) converges to a steady state solution and therefore its highest coefficient  $\hat{u}_N(t)$  is a bounded function of  $t$ . However the second term in (2.12) is not; in fact

$$\int_0^t \hat{g}_N(\tau) d\tau = t \hat{g}_N$$

and this diverges linearly in  $t$ .

The above does not contradict the usual stability argument since for fixed  $t$ ,  $u_N(x, t)$  converges to  $u(x, t)$  as  $N$  tends to infinity, but the usefulness of (2.6c) as a method for marching to steady state is doubtful. This will be demonstrated further in the numerical experiments.

**Proof:** We denote by  $P_N g(x, t)$  the (unique) interpolation polynomial of degree  $N$  in  $x$  that interpolates  $g(x, t)$  at the points  $x_l = \cos \frac{\pi l}{N}$   $0 \leq l \leq N$ . With this notation equations (2.6c)-(2.7) can be rewritten as

$$\frac{\partial u_N}{\partial t} = \frac{\partial^2 u_N}{\partial x^2} + P_N g - \frac{1}{2} \sum_{l=0}^N g(x_l) \alpha_l + (Ax + B) T'_N(x) \quad (2.14a)$$

$$u_N(x, 0) = 0 = \frac{\partial u_N}{\partial x}(1, t) = \frac{\partial u_N}{\partial x}(-1, t). \quad (2.14b)$$

By comparing the coefficients of the highest Chebyshev polynomial in both sides of (2.14a) one gets

$$A = \left[ \frac{d\hat{u}_N}{dt} - \hat{g}_N \right] \cdot \frac{1}{N}. \quad (2.15)$$

Multiplying now (2.14a) by  $\alpha_j$  and summing we get

$$\frac{d}{dt} \sum_{j=0}^N u_N(x_j, t) \alpha_j = \sum_{j=0}^N \frac{\partial^2 u_N}{\partial x^2}(x_j, t) \alpha_j + \sum_{j=0}^N g(x_j) \alpha_j - \sum_{l=0}^N g(x_l) \alpha_l + \sum_{j=0}^N (Ax_j + B) T'_N(x_j) \alpha_j \quad (2.16)$$

We use now the exactness of the Clenshaw Curtis formula and the fact that  $T'_N(x_j) = 0$   $1 \leq j \leq N-1$  to get

$$\begin{aligned} \frac{d}{dt} \sum_{j=0}^N u_N(x_j, t) \alpha_j &= \frac{\partial u_N}{\partial x}(1, t) - \frac{\partial u_N}{\partial x}(-1, t) + 2AN^2 \alpha_0 \\ &= \frac{2N^2}{N^2 - 1} \left[ \frac{d\hat{u}_N}{dt} - \hat{g}_N(t) \right] \cdot \frac{1}{N} \end{aligned} \quad (2.17)$$

Integration of (2.17) yields (2.12) and the proof is completed.

It is obvious that the problem lies in the use of the Clenshaw Curtis quadrature formula. Specifically, we need to use an accurate quadrature formula that does not use the boundary points. Such a formula is constructed in Lemma(1.3).

**Lemma (1.3):** Let  $f(x)$  be a polynomial of degree  $N-2$  at most. Let  $x_j = \cos \frac{\pi j}{N}$  then

$$\int_{-1}^1 f(x) dx = \sum_{j=1}^{N-1} g(x_j) \beta_j \quad (2.18a)$$

where

$$\beta_j = (-1)^{j+1} \frac{2}{N^2 - 1} + \alpha_j \quad j = 1, \dots, N-1 \quad (2.18b)$$

and  $\alpha_j$  are defined in (2.8).

**Proof:** Since  $f(x)$  is a polynomial of degree  $N-2$ , it is uniquely determined by its values at the interior points  $x_j, j = 1, \dots, N-1$ . In fact

$$f(x) = \sum_{j=1}^{N-1} f(x_j) \frac{T'_N(x)}{x - x_j} \cdot \frac{1}{T''_N(x_j)}. \quad (2.19)$$

Therefore

$$\int_{-1}^1 f(x) dx = \sum_{j=1}^{N-1} f(x_j) \beta_j$$

where

$$\beta_j = \int_{-1}^1 \frac{T'_N(x)}{x - x_j} \cdot \frac{1}{T''_N(x_j)} dx. \quad (2.20)$$

We use now the Clenshaw Curtis formula to evaluate the integral (2.20). Thus

$$\begin{aligned} \beta_j &= \sum_{\substack{l=0 \\ l \neq j}}^N \frac{T'_N(x_l)}{x_l - x_j} \frac{\alpha_l}{T''_N(x_j)} + \alpha_j \\ &= \frac{T'_N(1)}{(1 - x_j)T''_N(x_j)} \alpha_0 + \frac{T'_N(-1)\alpha_N}{(-1 - x_j)T''_N(x_j)} + \alpha_j \\ &= (-1)^{j+1} \frac{2}{N^2 - 1} + \alpha_j \end{aligned} \quad (2.21)$$

and the quadrature formula (2.18) is thus established.

Using Lemma (1.3) we suggest the following algorithm for the pseudospectral Chebyshev discretization of (2.1)-(2.2). In the new algorithm we seek  $u_N(x, t)$  such that

$$\frac{\partial u_N}{\partial t} = \frac{\partial^2 u_N}{\partial x^2} + g - \frac{1}{2} \sum_{j=1}^{N-1} g(x_j) \beta_j \quad \text{at } x = x_j \quad j = 1, \dots, N-1 \quad (2.22a)$$

$$\frac{\partial u_N}{\partial x}(1, t) = \frac{\partial u_N}{\partial x}(-1, t) \quad (2.22b)$$

$$u_N(x, 0) = 0. \quad (2.2c)$$

In the new method the compatibility condition (2.5) is satisfied. In fact we state

**Theorem (1.1):** Let  $u_N(x, t)$  be the solution of (2.22). Let  $\beta_j$  be defined in (2.21) then  $\sum_{j=1}^{N-1} u_N(x_j, t) \beta_j = 0$  for all  $t$ .

**Proof:** We multiply (2.22a) by  $\beta_j$  and sum to get

$$\frac{d}{dt} \sum_{j=1}^{N-1} u_N(x_j, t) \beta_j = \sum_{j=1}^{N-1} \frac{\partial^2 u_N}{\partial x^2}(x_j, t) \beta_j + \sum_{j=1}^{N-1} g(x_j) \beta_j - \sum_{j=1}^{N-1} g(x_j) \beta_j. \quad (2.23)$$

We use now Lemma (1.3) and the fact that  $\frac{\partial^2 u_N}{\partial x^2}$  is a polynomial of degree  $N - 2$ , to get

$$\sum_{j=1}^{N-1} \frac{\partial^2 u_N}{\partial x^2}(x_j, t) \beta_j = \int_{-1}^1 \frac{\partial^2 u_N}{\partial x^2} dx = \frac{\partial u_N}{\partial x}(1, t) - \frac{\partial u_N}{\partial x}(-1, t) = 0 \quad (2.24)$$

and therefore

$$\frac{d}{dt} \sum_{j=1}^{N-1} u_N(x_j, t) \beta_j = 0$$

and since  $u_N(x, 0) = 0$

$$\sum_{j=1}^{N-1} u_N(x_j, t) \beta_j = 0 \quad (2.25)$$

which proves the theorem.

### III. STEADY STATE PROBLEMS

In this section we discuss the equation

$$u_{xx} = g \quad (3.1a)$$

$$u_x(\pm 1) = 0 \quad (3.1b)$$

Equation (3.1) can be viewed as the steady state version of (2.1a), however this time we need one more condition to get a unique solution. We impose the condition

$$\int_{-1}^1 u \, dx = 0. \quad (3.2)$$

In order for (3.1a) and (3.1b) to be compatible  $g(x)$  has to satisfy the condition

$$\int_{-1}^1 g(x) dx = 0. \quad (3.3)$$

We have demonstrated in Section II that the use of the Clenshaw-Curtis quadrature formula (2.8) causes problems in solving (3.1) by trying to evolve the solution of (2.1) to steady state. Here we would like to discuss the direct solution of (3.1) via the influence matrix technique. We will demonstrate that the quadrature formula (2.18) developed in Section II should be used, rather than the Clenshaw-Curtis formula (2.8).

The solution of (3.1)-(3.2) via the influence matrix technique involves seeking the approximation of (3.1) as a sum of the solution of three problems:

**Problem 1:** Seek a polynomial  $w_N(x)$  such that

$$\begin{aligned} \frac{d^2 w_N}{dx^2} &= g - g_0 \text{ at } x = x_j = \cos \frac{\pi j}{N} \quad 1 \leq j \leq N \\ w_N(1) &= 0 = w_N(-1) \end{aligned} \quad (3.4)$$

and  $g_0$  is an approximation to  $\int g(x) dx$ . We will consider either

$$g_0 = \sum_{j=0}^N g(x_j) \alpha_j \quad (3.5a)$$

$$\alpha_j \text{ given in (2.8)}$$

or

$$g_0 = \sum_{j=1}^{N-1} g(x_j) \beta_j. \quad (3.5b)$$

$$\beta_j \text{ given in (2.18b)}$$

**Problem 2:** Seek a polynomial  $v^I(x)$  such that

$$\frac{d^2 v^I}{dx^2} = 0 \text{ at } x = x_j \quad j = 1, \dots, N-1 \quad (3.6a)$$

$$v^I(-1) = 0 \quad v^I(1) = 1 \quad (3.6b)$$

**Problem 3:** Seek a polynomial  $v^{II}(x)$  such that

$$\frac{d^2 v^{II}}{dx^2} = 0 \quad \text{at } x = x_j \quad j = 1, \dots, N-1 \quad (3.7a)$$

$$v^{II}(-1) = 1 \quad v^{II}(1) = 0 \quad (3.7b)$$

Given  $w_N, v^I, v^{II}$  we look for a solution of the form

$$u_N(x) = w_N(x) + d_1 v^I(x) + d_2 v^{II}(x). \quad (3.8)$$

Clearly  $u_N$  satisfies

$$\frac{d^2 u_N(x)}{dx^2} = g(x) - g_0 \quad \text{at } x = x_j \quad 1 \leq j \leq N$$

for any  $d_1, d_2$ . The constants  $d_1, d_2$  are determined by the boundary conditions (3.1b).

In our case (3.1b) (3.2) imply

$$0 = \frac{du_N}{dx}(-1) = \frac{dw_N(-1)}{dx} + d_1 \frac{dv^I}{dx}(-1) + d_2 \frac{dv^{II}}{dx}(-1) \quad (a)$$

$$0 = \frac{du_N(1)}{dx} = \frac{dw_N(1)}{dx} + d_1 \frac{dv^I(1)}{dx} + d_2 \frac{dv^{II}(1)}{dx} \quad (b)(3.9)$$

$$0 = \int_{-1}^1 u_N(x) dx = \int_{-1}^1 w_N(x) dx + d_1 \int_{-1}^1 v^I(x) dx + d_2 \int_{-1}^1 v^{II}(x) dx \quad (c)$$

Ideally (3.9a) and (3.9b) should yield the same equation for  $d_1, d_2$ . The integrals in (3.9c) have, of course, to be discretized.

In this simple model problem we are able to evaluate explicitly  $w_N(x), v^I(x)$ , and  $v^{II}(x)$ . In fact we can state

**Lemma (2.1):** The solutions of problems 2 and 3 are given by

$$v^I = \frac{1+x}{2} \quad v^{II} = \frac{1-x}{2} \quad (3.10)$$

**Proof:**  $v^I(x), v^{II}(x)$  in (3.10) are the only polynomials that satisfy (3.6), (3.7) respectively.

In order to get an expression for  $w_N(x)$  we introduce the interpolation polynomial  $P_{N-2g}$  that interpolates  $g(x)$  at the points  $x_j \quad 1 \leq j \leq N-1$ . By (2.19)

$$g(x) = \sum_{j=1}^{N-1} g(x_j) \frac{T'_N(x)}{x - x_j} \cdot \frac{1}{T''_N(x_j)}. \quad (3.11)$$

It is clear from (3.4) that  $w_N(x)$  satisfies

$$\frac{d^2 w_N}{dx^2} = P_{N-2}g - g_0 \quad (3.12a)$$

$$w_N(1) = 0 = w_N(-1). \quad (3.12b)$$

Note that equation (3.12a) holds for *any*  $-1 \leq x \leq 1$ , not just for the grid points  $x_j$ ; it is therefore possible to integrate (3.12a) to get

$$\frac{dw_N}{dx}(x) = \frac{dw_N}{dx}(-1) + \int_{-1}^x (P_{N-2}g(\xi) - g_0) d\xi \quad (3.13)$$

and

$$w_N(x) = \frac{dw_N}{dx}(-1)(1+x) + \int_{-1}^x (x-\eta)[P_{N-2}g(\eta) - g_0] d\eta. \quad (3.14)$$

Substituting now the condition  $w_N(1) = 0$  we get

$$\frac{dw_N}{dx}(-1) = -\frac{1}{2} \int_{-1}^1 (1-\eta)[P_{N-2}g(\eta) - g_0] d\eta. \quad (3.15)$$

We are ready now for the main result of this section.

**Lemma (2.2):** Let  $g_0$  be defined by the use of the Clenshaw-Curtis formula as in (3.5a), then equation (3.9a) is incompatible with (3.9b).

**Proof:** Using the expressions (3.10) for  $v^I$  and  $v^{II}$  in (3.9a,b) we get

$$\frac{dw_N}{dx}(-1) = \frac{d_1}{2} - \frac{d_2}{2} \quad (a)$$

$$(3.16)$$

$$\frac{dw_N}{dx}(1) = \frac{d_1}{2} - \frac{d_2}{2}. \quad (b)$$

However from (3.13)

$$\frac{dw_N}{dx}(1) = \frac{dw_N}{dx}(-1) + \int_{-1}^1 [P_{N-2}g(\xi) - g_0] d\xi. \quad (3.17)$$

Since  $P_{N-2}g$  is a polynomial of degree  $N-2$ , the quadrature formula (2.18) is exact and therefore

$$\begin{aligned} \int_{-1}^1 [P_{N-2}g(\xi) - g_0] d\xi &= \sum_{j=1}^N g(x_j) \beta_j - \sum_{j=0}^N g(x_j) \alpha_j \\ &= -g(x_0) \alpha_0 + g(x_N) \alpha_N - \sum_{j=0}^N (-1)^j \frac{2}{N^2-1} g(x_j) \\ &= -2 \sum_{j=0}^N \frac{(-1)^j g(x_j)}{c_j} \end{aligned} \quad (3.18)$$

where  $c_0 = 2 = c_N$ ,  $c_j = 1$   
so that

$$\frac{dw_N}{dx}(1) \neq \frac{dw_N}{dx}(-1) \quad (3.19)$$

which proves the Lemma.

However if one uses the new quadrature formula as in (3.5b) one gets

**Lemma (2.3):** Let  $g_0$  be defined in (3.5a), then (3.9)(a) and (3.9)(b) are compatible.

**Proof:** We start as in the last Lemma except now

$$\int_{-1}^1 (P_{N-2}g - g_0)d\xi = \sum_{j=1}^N g(x_j)\beta_j - \sum_{j=1}^N g(x_j)\beta_j = 0$$

which yields the result.

We have thus established that the use of the Clenshaw-Curtis formula may lead to an approximation that does not satisfy (3.1b). The new quadrature formula alleviates this problem.



## IV. NUMERICAL RESULTS

In Section II we showed that the use of the Clenshaw-Curtis formula (2.8),(2.9) may cause problems when one attempts to solve the steady state problem corresponding to (2.1). In this Section we report on the numerical solution of equation (2.1), (2.2) with two different source functions,  $g(x) = \cos 3\pi x$  and  $g(x) = \cos 5\pi x$ . To advance in time we use the fourth order Runge-Kutta scheme; a grid of 19 points was used. In table I we summarize the results for the first case:  $g(x) = \cos 3\pi x$ . The first column gives the  $L_2$  deviation from the steady state solution when (2.6a,b)-(2.7) is used, i.e. no modification for the right hand side. A linear growth is observed in the deviation from the steady state. The solution did not converge even after 200,000 time steps.

In the second column we give the results for (2.6c), in which the approximation to the mean of  $g$  is obtained with the use of the Clenshaw-Curtiss formula. It is interesting to note that the results are essentially the same; the subtraction did not improve the situation. In the third column we present the results for (2.22a); convergence was obtained after 10,000 time steps.

In table II we present the same results for  $g(x) = \cos 5\pi x$ . Basically, the results are the same as in table I; however, the divergence is much more rapid than in the first case. The new method converged after 15,000 time steps.

Table I  
 $L_2$  Errors for  $g(x) = \cos 3\pi x$   $N = 19$

No. Of Time Steps	No Subtraction	Clenshaw-Curtiss Subtraction	GS Subtraction
0	1.778 (-3)	1.778 (-3)	1.778 (-3)
10000	1.589 (-6)	8.644 (-7)	2.669 (-7)
15000	2.324 (-6)	2.32 (-6)	2.669 (-7)*
100000	1.492 (-5)	1.494 (-5)	same
150000	2.234 (-5)	2.237 (-5)	same
200000	2.3013 (-5)	2.9056 (-5)	same
Growth	7.4165 (-7)	7.4277 (-7)	0

\* converges after 25000 steps

Table II  
 $L_2$  Errors for  $g(x) = \cos 5\pi x$   $N = 19$

No. Of Time Steps	No Subtraction	Clenshaw-Curtiss Subtraction	GS Subtraction
0	6.320 (-4)	6.320 (-4)	6.320 (-4)
10000	1.4509 (-3)	1.456 (-3)	2.4405 (-4)
15000	2.1173 (-3)	2.126 (-3)	2.4554 (-4)
100000	1.3511 (-2)	1.3606 (-2)	same
150000	2 (-2)	2.036 (-2)	same
200000	2.581 (-2)	2.644 (-2)	same
Growth	6.46 (-4)	6.75 (-4)	0

Finally, we show the results of a two-dimensional calculation for:

$$u_{xx} + u_{yy} = g(x, y) = \cos \pi x \cdot \cos \pi y \quad (4.1a)$$

with

$$u_x = 0 \text{ on } (\pm 1, y) \quad (4.1b)$$

$$u_y = 0 \text{ on } (x, \pm 1). \quad (4.1c)$$

Solution is carried out by the influence matrix technique, with the related Dirichlet Poisson problems solved via the direct solution method of tensor-product diagonalization. In order to set the level of the solution, that is, to set the arbitrary constant in the solution of the pure Neumann problem, one must set the value of the solution at one point on the boundary by replacing one row of the influence matrix. However, if the source function does not satisfy the discrete version of the compatibility relation

$$\int_{-1}^1 \int_{-1}^1 g(x, y) dx dy = 0 \quad (4.2)$$

then large distortions appear in the solution near that point. On the other hand, if (4.2) is discretely satisfied, then the solution near the set point will be regular.

In Figure 1 is shown isolines of the solution to (4.1), for which  $g(x, y)$  was replaced by

$$g(x, y) = \sum_{l=0}^{N_x} \sum_{k=0}^{N_y} g(x_l, y_k) \alpha_k \alpha_l \quad (4.3)$$

that is, using the Clenshaw-Curtiss formula to satisfy compatibility. The location of the boundary point which is set in the influence matrix is obvious; the true solution is swamped by the artifact distortions. In Figure 2 is shown the isolines for the solution wherein the compatibility relation is satisfied using the quadrature formula developed here. The solution is now smooth, symmetric, and regular near the set point.

## References

- [1] Canuto, C., Hussaini, M. Y., Quarteroni, A., and Zang, T. A., *Spectral Methods in Fluid Dynamics*, Springer-Verlag, N.Y., 1987.
- [2] Davis, P. J., and Rabinowitz, P., *Methods of Numerical Integration*, 2nd Edition, Academic Press, N.Y., 1984.
- [3] Gottlieb, D., Hussaini, M. Y., and Orszag, S. A., "Theory and Applications of Spectral Methods", in *Spectral Methods for Partial Differential Equations*, ED. by R. G. Voigt, D. Gottlieb, M. Y. Hussaini, SIAM-CBMS, pp. 1-54.
- [4] Orszag, S. A., Israeli, M., and Deville, M. O., "Boundary Conditions for Incompressible Flows", *J. Sci. Comp.*, Vol. 1, 1986, pp. 75-111.
- [5] Streett, C. L., and Macaraeg, M. G., "Spectral Multidomain for Large-Scale Fluid Dynamic Simulations", *Appl. Num. Math.*, Vol. 6, 1989, pp. 123-139.

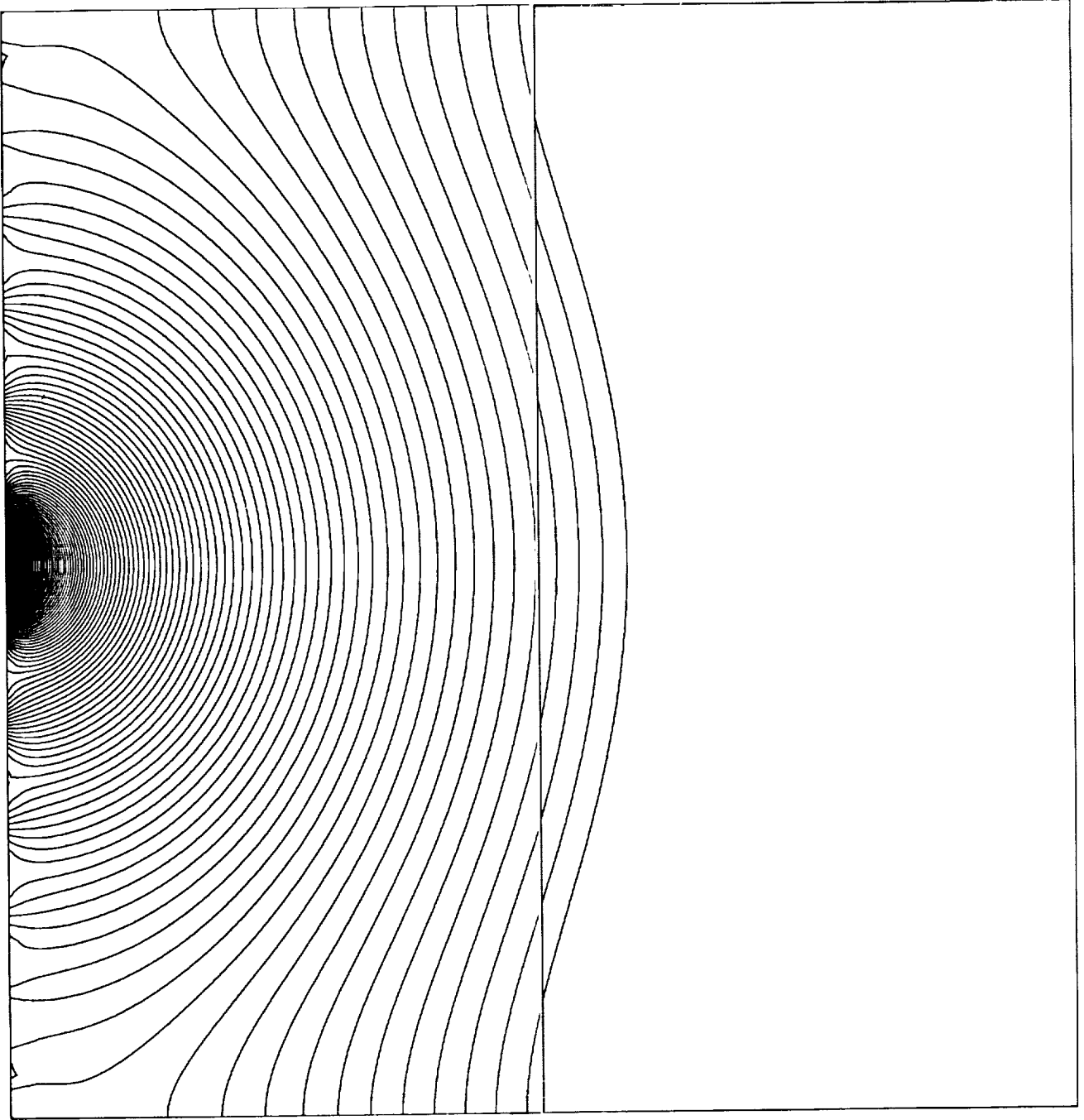


Fig. 1. Isolines of solution to (4.1) with Clenshaw-Curtis subtraction.

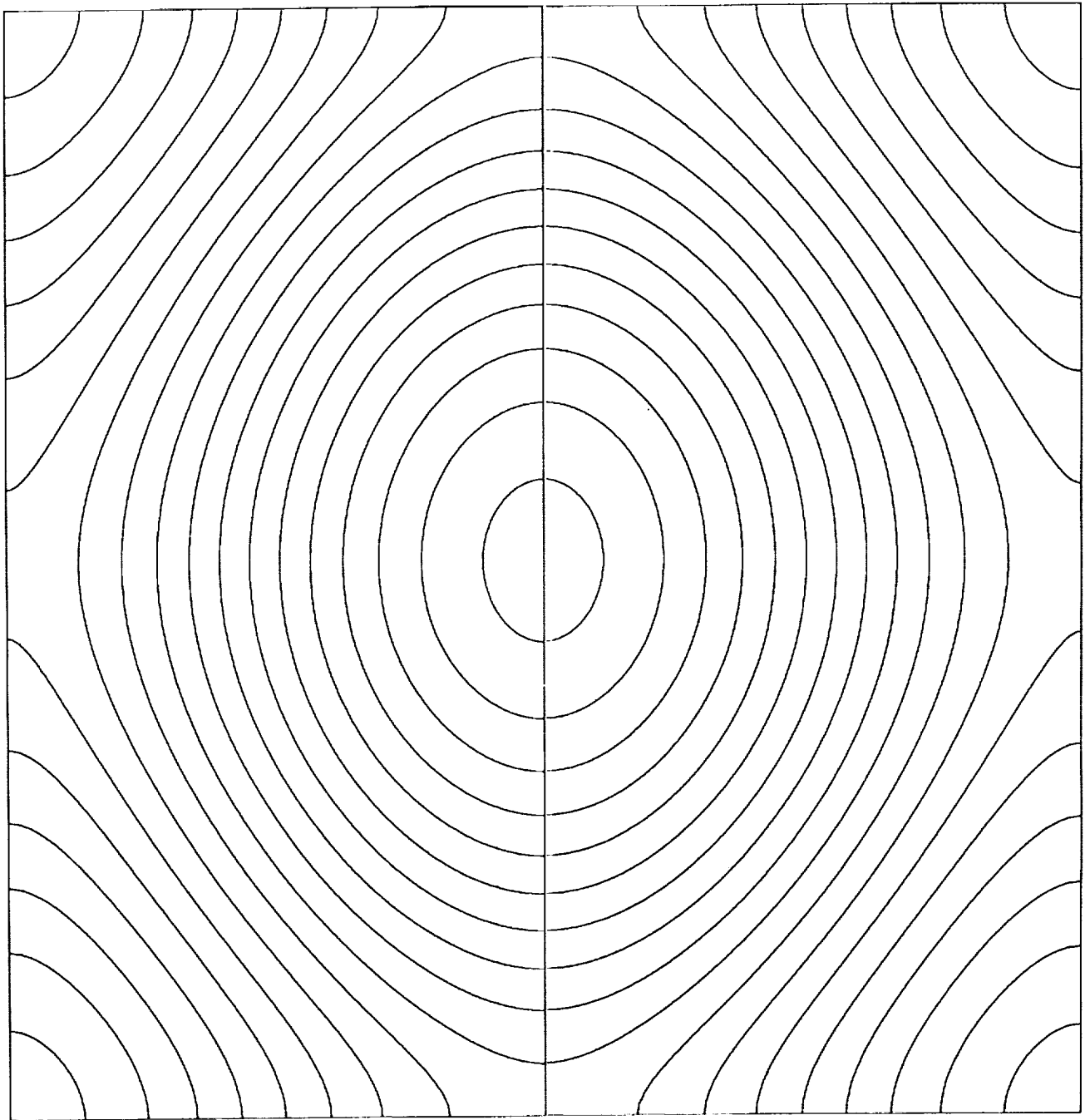


Fig. 2. Isolines of solution to (4.1) with GS subtraction.









## Report Documentation Page

1. Report No. NASA CR-187458 ICASE Report No. 90-74		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  QUADRATURE IMPOSITION OF COMPATABILITY CONDITIONS IN CHEBYSHEV METHODS				5. Report Date October 1990	
				6. Performing Organization Code	
7. Author(s)  D. Gottlieb C. L. Streett				8. Performing Organization Report No.  90-74	
				10. Work Unit No.  505-90-21-01	
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225				11. Contract or Grant No. NAS1-18107 NAS1-18605	
				13. Type of Report and Period Covered  Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225				14. Sponsoring Agency Code	
15. Supplementary Notes Langley Technical Monitor: Richard W. Barnwell  Submitted to Journal of Scientific Computing  Final Report					
16. Abstract <p>Often, in solving an elliptic equation with Neumann boundary conditions, a compatability condition has to be imposed for well-posedness. This condition involves integrals of the forcing function.</p> <p>When pseudospectral Chebyshev methods are used to discretize the partial differential equation, these integrals have to be approximated by an appropriate quadrature formula. The Gauss-Chebyshev (or any variant of it, like the Gauss-Lobatto) formula can not be used here since the integrals under consideration do not include the weight function. A natural candidate to be used in approximating the integrals is the Clenshaw-Curtis formula, however we show in this paper that this is the wrong choice and it may lead to divergence if time dependent methods are used to march the solution to steady state.</p> <p>We develop, in this paper, the correct quadrature formula for these problems. This formula takes into account the degree of the polynomials involved. We show that this formula leads to a well conditioned Chebyshev approximation to the differential equations and that the compatability condition is automatically satisfied.</p>					
17. Key Words (Suggested by Author(s))  quadrature formulas, compatability condition, Chebyshev methods			18. Distribution Statement 34 - Fluid Mechanics and Heat Transfer 64 - Numerical Analysis  Unclassified - Unlimited		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified		21. No. of pages 20	22. Price A03	

