Annals of Mathematical Sciences and Applications
Volume 5, Number 1, 101–138, 2020

High accuracy compact difference schemes for differential equations in mathematical sciences

Murli M. Gupta*,†

Our work on high order compact difference schemes was initiated about 35 years ago when we first presented new 4th and 6th order discretizations for convection-diffusion equations in 2-dimensions. This work is now routinely applied to complex fluid flow problems, and has also been developed for 3-dimensional differential equations. In our quest to apply these ideas to the biharmonic equation, we discovered that it is beneficial to carry the unknowns and their derivatives as computational parameters. This allowed us to propose the streamfunction-velocity formulation for the Navier–Stokes equations.

In this paper, I describe the historical developments of the high order compact difference schemes and their evolution into the powerful computational techniques that are now available to solve fluid flow problems of important physical interest. Some theoretical analysis on stability and convergence of these schemes will also be presented.

This paper is based, in part, on the presentation I gave in March 2019 at the Taiwan-India Joint Conference – Recent Progress on Flow Simulation and Stability Analysis: 2019 Spring Progress in Mathematical and Computational Studies on Science and Engineering Problems at National Taiwan University, Taipei, Taiwan.

AMS 2000 subject classifications: Primary 65N06, 65M06; Secondary 76M20.
Keywords and phrases: High order compact, Navier–Stokes, streamfunction-velocity formulation, high-accuracy, fluid flow.

1. Introduction

High order compact schemes are ubiquitous now. We started working on the development of such schemes in the early 1980s with the initial goal of solving the convection-diffusion equations with high degree of accuracy. At that

*Corresponding author.
†ORCID: 0000-0003-1055-8166.

time, we had a choice of two methods: the central difference scheme (CDS) and the upwind difference scheme (UDS). The central difference scheme has a truncation error of second order but it only worked when the convection terms were of moderate size. In the case of high convection, CDS often failed to converge and, in any case, gave incorrect, oscillatory solutions. The upwind difference scheme (with truncation error of first order) was able to converge in most cases but the solutions were not as accurate and the issues of artificial diffusion arose in many situations.

We started working on the convection-diffusion equations with the eventual goal of working on the Navier–Stokes equations which have many practical applications. We proposed fourth order compact schemes for the convection-diffusion equations and showed that these new schemes were successful in converging in all cases, and provided high accuracy solutions of all test problems. Subsequently, we were able to provide rigorous proofs of the stability and convergence of many of these schemes. Once this was done, we decided to work on the biharmonic equations which represent Stokes flow and are linearized form of the Navier–Stokes equations. Discretization of the biharmonic equation traditionally required a 13-point stencil which needed modifications near the boundaries of the physical domain. We proposed a compact finite difference approximation for the two-dimensional biharmonic equation that required only 9 grid points in a compact square grid cell; this approximation carried the unknown solution and its first order derivatives as parameters for the computational procedure. We were able to show that our new formulation for the 2D biharmonic equation has a truncation error of second order on the 9 point stencil. We then obtained fourth order finite difference approximations for the two dimensional biharmonic equation still on a 9 point stencil; subsequently we also obtained compact high order approximations for the three dimensional biharmonic equation on a unit cube. Second order approximations were obtained on a 19 point stencil for 3D biharmonic equations while 27 point stencils were used to obtain 4th order accurate approximations. In each of these cases, we used the unknown solution and its first derivatives as the variables to carry as computational parameters in the computational process. This gave rise to a new paradigm of streamfunction-velocity formulations for the Navier–Stokes equations.

Once we were successful with the biharmonic equation, we worked on the steady state Navier–Stokes equations in two-dimensions and proposed a streamfunction-velocity formulation. This formulation obviated the need for dealing with the pressure equation (in the case of primitive variable formulation) and the vorticity equation with unknown boundary values (in the case

of streamfunction-vorticity formulation). Even though fourth order approximations are available for Navier–Stokes equations, we have decided to only use the second order approximations due to the complexity of the differential equations. Once the steady state problems were tackled satisfactorily, we turned our attention to the time-dependent Navier–Stokes equations and discovered that our formulations provided high accuracy solutions in the case of fluid flow problems of high complexity.

This paper is arranged as follows:

2. Early years of HOC scheme

2.1. Poisson equation

We consider 2- and 3-dimensional Poisson equations:

$$\triangle u = u_{xx} + u_{yy} = f \qquad (2D)$$
$$\triangle u = u_{xx} + u_{yy} + u_{zz} = f \quad (3D)$$

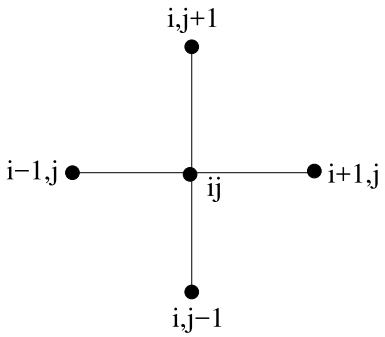

Figure 1: The standard five point stencil for the 2D Poisson equation.

Standard 5 point approximation at a grid point (x_i, y_j) in 2D is given by

$$-4u_{i,j} + u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1} = h^2 f_{i,j}$$

Here the subscripts i, j indicate values of the function u at a particular grid point (x_i, y_j). This formula utilizes the function values of the four nearest neighbors of the i, j point and has a truncation error of order h^2 (Figure 1).

The standard 9 point formula in 2D (with truncation error of order h^4) is given by

$$\begin{aligned} &-60u_{i,j} + 16(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1}) \\ &\quad -(u_{i-2,j} + u_{i+2,j} + u_{i,j+2} + u_{i,j-2}) = 12h^2 f_{i,j} \end{aligned}$$

This formula uses points that are two grid units away from the central point (x_i, y_j) which causes large bandwidth and difficulties near the boundaries (Figure 2).

A Compact Fourth Order Formula for 2D Poisson Equation is given by

$$\begin{aligned} &4(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1}) \\ &\quad + (u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i+1,j-1}) - 20u_{i,j} \\ &= 0.5h^2(8f_{i,j} + f_{i,j-1} + f_{i-1,j} + f_{i+1,j} + f_{i,j+1}) \end{aligned}$$

The above method has a truncation error of order $O(h^4)$; it uses only 8 nearest neighbours of the central point (x_i, y_j) (Figure 3) and was called Mehrstellenverfahren (also called Hermitian or many point method) by L. Collatz (1951) [12].

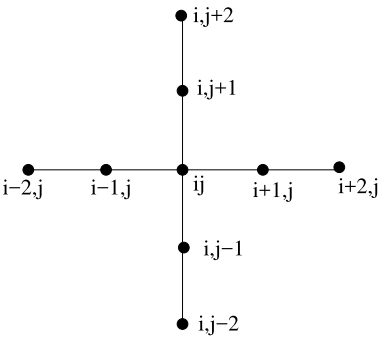

Figure 2: The standard nine point stencil for the 2D Poisson equation.

In a ground breaking work [22], we compared the accuracy and computational efficiency of the above difference approximations for some test problems. We found that while Mehrstellen formula indeed provided fourth-order-accurate values of the unknowns, the values of first derivatives were still only of second order accuracy because only the second order accurate approximations (central difference schemes) were available to compute the first derivatives of the unknowns at that time. We introduced, in 1984, new finite difference approximations for computing the numerical values of the first derivatives of $u(x, y)$ (i.e., $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$) in [22] and found that these approximations yielded $O(h^4)$ accuracy for the first derivatives when used in conjunction with the nine-point Mehrstellen formula. These approximations were obtained in the process of obtaining the high order compact schemes for the convection-diffusion equations (see next section).

2.2. Convection-diffusion equation

$$\triangle u + pu_x + qu_y = f$$

Our initial work was done with Professors Ram P. Manohar and John W. Stephenson at the University of Saskatchewan, Saskatoon, Canada, in the early 1980s [36]. The seminal paper on this topic dealt with the convection-diffusion equation with constant coefficients (p and q are constant) and utilized the compact stencil (Figure 3).

This work was motivated by a desire to obtain high accuracy solutions of these problems in cases when convection was dominant (i.e., the convection coefficients p and q are large). Prior to our work, there were two typical methods to discretize these differential equations: upwind difference scheme (UDS) and central difference scheme (CDS). Upwind difference schemes

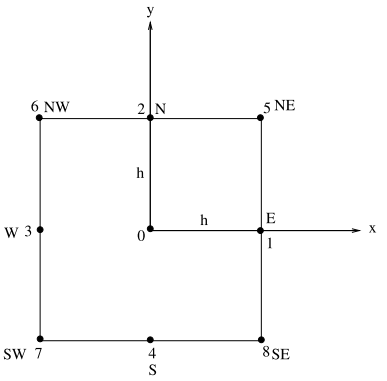

Figure 3: The 9 point compact stencil for the 2D convection diffusion equation.

(UDS) were of first order of accuracy (truncation error of $O(h)$) and provided a convergent solution process. However, the solutions were not very accurate and artificial diffusion was often prevalent when convection was dominant. The Central difference schemes (CDS) were the work horse of the time and provided $O(h^2)$ accuracy. However, this became problematic when the convection terms were dominant. In this case, the iterative methods often failed to converge; even when the solutions were obtained (e.g., using direct solvers), the numerical solutions exhibited nonphysical oscillations.

Use of high order approximations at that time required solution values at points that were two grid points away from the central point (Figure 2). This required the development of modified finite difference approximations for grid points close to boundaries.

The work on high order compact (HOC) schemes was initiated in collaboration with Professors Manohar and Stephenson of the University of Saskatchewan, Saskatoon, Canada. We initially carried out these derivations using reams of pen and paper work; in later years we were able to reproduce, and extend, many of these derivations using symbolic algebra packages such as Mathematica.

The first announcement of our results was made by the author at the Second National Symposium on The Numerical Methods in Heat Transfer held at University of Maryland, College Park in September 1981 [36]. These works were then presented by me and the two coauthors (Manohar and Stephenson) at the conference of American Society of Mechanical Engineers in Cincinnati, Ohio (1981), 10th IMACS World Congress on System Simulation and Scientific Computation in Montreal, Canada (1982), International Symposium on Refined Modelling of Flows in Paris, France (1982), Fourth

International Symposium on Finite Element Methods in Flow Problems in Tokyo, Japan (1982), and Canadian Congress of Applied Mechanics (CANCAM) in Saskatoon, Canada (1983). Some of the presented results were published in the proceedings of these conferences. I note that the results obtained at that time were computed on computer hardware that could only deal with grid sizes as fine as the 40×40 grid with 1681 unknowns. Later on, with much improvements in computer hardware and software, we were able to carry out such computations on much finer grids as well as in three dimensions.

Initially, our work on convection-diffusion equations centered on the constant coefficient problems. Once this was successful, we focused on variable coefficient problems and presented the new formulation in Scientific Computing (1983) [20] and IJNMF (1984) [37]. A further generalization for two dimensional elliptic equations with variable coefficients was published in the inaugural volume of NMPDE (1985) [38].

The fourth order compact approximation for the convection-diffusion equation with variable coefficients ([37], IJNMF, 1984) is given below. The constant coefficient approximations obtained by us initially can be obtained by simply turning the variables $p(x,y)$ and $q(x,y)$ into constants.

$$Lu \equiv u_{xx} + u_{yy} + p(x,y)u_x + q(x,y)u_y = f(x,y)$$

The magnitudes of $p(x,y)$ and $q(x,y)$ determine the ratio of the convection to diffusion; the finite difference approximation is given by [37]:

$$\sum_{j=0}^{8} \alpha_j u_j = \frac{h^2}{2}[f_N + f_S + f_E + f_W + 8f_0] + \frac{h^3}{4}[p_0(f_E - f_W) + q_0(f_N - f_S)]$$

where

$$\begin{aligned}
\alpha_1 = \alpha_E &= 4 + \frac{h}{4}[4p_0 + 3p_E - p_W + p_N + p_S] \\
&\quad + \frac{h^2}{8}[4p_0^2 + p_0(p_E - p_W) + q_0(p_N - p_S)] \\
\alpha_2 = \alpha_N &= 4 + \frac{h}{4}[4q_0 + 3q_N - q_S + q_E + q_W] \\
&\quad + \frac{h^2}{8}[4q_0^2 + p_0(q_E - q_W) + q_0(q_N - q_S)] \\
\alpha_3 = \alpha_W &= 4 - \frac{h}{4}[4p_0 - p_E + 3p_W + p_N + p_S] \\
&\quad + \frac{h^2}{8}[4p_0^2 - p_0(p_E - p_W) - q_0(p_N - p_S)]
\end{aligned}$$

$$\alpha_4 = \alpha_S = 4 - \frac{h}{4}[4q_0 - q_N + 3q_S + q_E + q_W]$$
$$+ \frac{h^2}{8}[4q_0^2 - p_0(q_E - q_W) - q_0(q_N - q_S)]$$
$$\alpha_5 = \alpha_{NE} = 1 + \frac{h}{2}[p_0 + q_0] + R_7$$
$$\alpha_6 = \alpha_{NW} = 1 - \frac{h}{2}[p_0 - q_0] - R_7$$
$$\alpha_7 = \alpha_{SW} = 1 - \frac{h}{2}[p_0 + q_0] + R_7$$
$$\alpha_8 = \alpha_{SE} = 1 + \frac{h}{2}[p_0 - q_0] - R_7$$
$$R_7 = \frac{h}{8}[q_E - q_W + p_N - p_S] + \frac{h^2}{4}p_0 q_0$$
$$\alpha_0 = -[20 + h^2(p_0^2 + q_0^2) + h(p_E - p_W) + h(q_N - q_S)]$$

2.3. Poisson and convection-diffusion equation: further progress

In the 1990's, with the help of my graduate students, Jun Zhang and Jules Kouatchou, we initiated further work on the convection-diffusion equations. We applied multigrid techniques to the two dimensional problems and obtained high levels of accuracy and grid-independent convergence rates.

In [28], we developed an efficient solution procedure for high accuracy solution of the convection-diffusion equation $\triangle u + pu_x + qu_y = f$ in conjunction with multigrid method. Our algorithm displayed grid independent convergence rates and provided solutions with high accuracy. Note that we were able to use much finer grids at that time due to the dramatic improvements in computer hardware and software.

Around this time Spotz and Carey published their work on higher order compact schemes which were similar to our schemes from a decade earlier, though their derivations were different [61, 60, 59]. Lele [48] also contributed to these derivations.

We carried out detailed comparisons of second and fourth order discretizations of Poisson equation using multigrid techniques [29] and showed that our method achieved dramatic improvements in computer efficiency and accuracy on both serial (SUN SPARCStation) and vector (Cray C90) computers. Note that at this point, we were able to work with grids as small as 256×256 with 66049 unknowns.

We also worked on the convection-diffusion equation in 3 dimensions

$$\triangle u + \lambda u_x + \mu u_y + \omega u_z = f$$

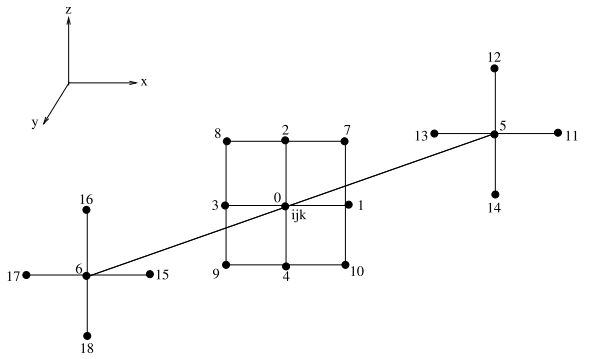

Figure 4: The 19 point stencil for the 3D convection diffusion equation.

In [39], we presented an explicit fourth-order compact finite difference scheme for approximating the three-dimensional (3D) convection-diffusion equation with variable coefficients. This 19-point formula, given below, is defined on a uniform cubic grid (Figure 4).

The fourth order approximation of above equation is given by

$$\sum_{l=0}^{18} c_l u_l = F_0$$

where the coefficients c_l are given by

$$\begin{aligned}
c_0 &= -[24 + h^2(\lambda_0^2 + \mu_0^2 + \omega_0^2) + h(\lambda_1 - \lambda_3 + \mu_2 - \mu_4 + \omega_5 - \omega_6)] \\
c_1 &= 2 - \frac{h}{4}(2\lambda_0 - 3\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4 - \lambda_5 - \lambda_6) \\
&\quad + \frac{h^2}{8}[4\lambda_0^2 + \lambda_0(\lambda_1 - \lambda_3) + \mu_0(\lambda_2 - \lambda_4) + \omega_0(\lambda_5 - \lambda_6)] \\
c_2 &= 2 - \frac{h}{4}(2\mu_0 - \mu_1 - 3\mu_2 - \mu_3 + \mu_4 - \mu_5 - \mu_6) \\
&\quad + \frac{h^2}{8}[4\mu_0^2 + \lambda_0(\mu_1 - \mu_3) + \mu_0(\mu_2 - \mu_4) + \omega_0(\mu_5 - \mu_6)] \\
c_3 &= 2 + \frac{h}{4}(2\lambda_0 + \lambda_1 - \lambda_2 - 3\lambda_3 - \lambda_4 - \lambda_5 - \lambda_6) \\
&\quad + \frac{h^2}{8}[4\lambda_0^2 - \lambda_0(\lambda_1 - \lambda_3) - \mu_0(\lambda_2 - \lambda_4) - \omega_0(\lambda_5 - \lambda_6)] \\
c_4 &= 2 + \frac{h}{4}(2\mu_0 - \mu_1 + \mu_2 - \mu_3 - 3\mu_4 - \mu_5 - \mu_6)
\end{aligned}$$

$$+\frac{h^2}{8}[4\mu_0^2-\lambda_0(\mu_1-\mu_3)-\mu_0(\mu_2-\mu_4)-\omega_0(\mu_5-\mu_6)]$$

$$c_5=2-\frac{h}{4}(2\omega_0-\omega_1-\omega_2-\omega_3-\omega_4-3\omega_5+\omega_6)$$
$$+\frac{h^2}{8}[4\omega_0^2+\lambda_0(\omega_1-\omega_3)+\mu_0(\omega_2-\omega_4)+\omega_0(\omega_5-\omega_6)]$$

$$c_6=2+\frac{h}{4}(2\omega_0-\omega_1-\omega_2-\omega_3-\omega_4+\omega_5-3\omega_6)$$
$$+\frac{h^2}{8}[4\omega_0^2-\lambda_0(\omega_1-\omega_3)-\mu_0(\omega_2-\omega_4)-\omega_0(\omega_5-\omega_6)]$$

$$c_7=1+\frac{h}{2}(\lambda_0+\mu_0)+\frac{h}{8}[\lambda_2-\lambda_4+\mu_1-\mu_3]+\frac{h^2}{4}\lambda_0\mu_0$$

$$c_8=1-\frac{h}{2}(\lambda_0-\mu_0)-\frac{h}{8}[\lambda_2-\lambda_4+\mu_1-\mu_3]-\frac{h^2}{4}\lambda_0\mu_0$$

$$c_9=1-\frac{h}{2}(\lambda_0+\mu_0)+\frac{h}{8}[\lambda_2-\lambda_4+\mu_1-\mu_3]+\frac{h^2}{4}\lambda_0\mu_0$$

$$c_{10}=1+\frac{h}{2}(\lambda_0-\mu_0)-\frac{h}{8}[\lambda_2-\lambda_4+\mu_1-\mu_3]-\frac{h^2}{4}\lambda_0\mu_0$$

$$c_{11}=1+\frac{h}{2}(\lambda_0+\omega_0)+\frac{h}{8}[\lambda_5-\lambda_6+\omega_1-\omega_3]+\frac{h^2}{4}\lambda_0\omega_0$$

$$c_{12}=1+\frac{h}{2}(\mu_0+\omega_0)+\frac{h}{8}[\mu_5-\mu_6+\omega_2-\omega_4]+\frac{h^2}{4}\mu_0\omega_0$$

$$c_{13}=1-\frac{h}{2}(\lambda_0-\omega_0)-\frac{h}{8}[\lambda_5-\lambda_6+\omega_1-\omega_3]-\frac{h^2}{4}\lambda_0\omega_0$$

$$c_{14}=1-\frac{h}{2}(\mu_0-\omega_0)-\frac{h}{8}[\mu_5-\mu_6+\omega_2-\omega_4]-\frac{h^2}{4}\mu_0\omega_0$$

$$c_{15}=1+\frac{h}{2}(\lambda_0-\omega_0)-\frac{h}{8}[\lambda_5-\lambda_6+\omega_1-\omega_3]-\frac{h^2}{4}\lambda_0\omega_0$$

$$c_{16}=1+\frac{h}{2}(\mu_0-\omega_0)-\frac{h}{8}[\mu_5-\mu_6+\omega_2-\omega_4]-\frac{h^2}{4}\mu_0\omega_0$$

$$c_{17}=1-\frac{h}{2}(\lambda_0+\omega_0)+\frac{h}{8}[\lambda_5-\lambda_6+\omega_1-\omega_3]+\frac{h^2}{4}\lambda_0\omega_0$$

$$c_{18}=1-\frac{h}{2}(\mu_0+\omega_0)+\frac{h}{8}[\mu_5-\mu_6+\omega_2-\omega_4]+\frac{h^2}{4}\mu_0\omega_0$$

$$F_0=\frac{h^2}{4}(6f_0+f_1+f_2+f_3+f_4+f_5+f_6)$$
$$+\frac{h^3}{4}[\lambda_0(f_1-f_3)+\mu_0(f_2-f_4)+\omega_0(f_5-f_6)].$$

When $\lambda = \mu = \omega \equiv 0$, the differential equation reduces to the $3D$ Poisson equation and this finite difference scheme reduces to the 19-point formula for that equation, see [27].

2.4. Convergence results

Proof of convergence of this 19-point finite difference approximation for 3D convection-diffusion equation with constant coefficients was obtained by us in collaboration with researchers from the Republic of Georgia (Professors Givi Berikelashvili and Manana Mirianashvili). This work was published in SIAM Journal on Numerical Analysis in 2007 [10] and contained the following main result:

Theorem 2.1. *Let the exact solution of the boundary value problem belong to $W_2^s(\Omega)$, $s > 1.5$. Then the discretization error of the finite difference scheme in the discrete $W_2^m(\omega)$-norm is $O(h^{s-m})$ where the parameter s satisfies the condition $max(1.5, m) < s \leq m + 4$, $m = 0, 1, 2$.*

This result shows that the convergence rate can be as high as $O(h^4)$ depending upon the smoothness of the exact solution of the underlying problem.

Further work (with Berikelashvili and Bidzina Midodashvili) on the 3D convection-diffusion equation with variable coefficients was presented at ICNAAM 2014 conference in Rhodes, Greece (September 2014) and appeared in AIP Conference Proceedings (2015) [9]. This work proved that our method for variable coefficient problems converges in the discrete L_2-norm with the rate h^m when the solution of the original problem belongs to the Sobolev space W_2^m, $2 < m \leq 4$.

A more recent result (with Berikelashvili and Midodashvili) concerns 3D Poisson equation with nonlocal boundary conditions and was published in Mathematical Sciences Letters (2018) [8]. The main theorem again provides convergence results of order up to h^4:

Theorem 2.2. *Let the solution of the Poisson equation belong to the space $W_2^s(\Omega)$, $s \geq 2$. Then the convergence rate of the corrected difference scheme in the discrete L_2-norm is defined by the estimate*

$$||U - u||_{L_2(\omega;r)} \leq ch^s ||u||_{W_2^s(\Omega)}, 2 \leq s \leq 4.$$

2.5. Scheme derivations using software

In [27], we published a variety of Mathematica codes for 3D Poisson equation. An example of such a code using 19 grid points in a unit box is presented below.

```
Clear[p,q,i,j,l,k,xx,yy,zz,u,eq,equ,a,b];

   (* Form the polynomial p of degree n *)

n=6;
p=0;
Clear[i,j,l,a];
Do[If[i+j+l<=n,p=p+a[i,j,l] x^i y^j z^l,Continue[]],{i,0,n},{j,0,n},{l,0,n}];

   (* Define the coordinates of the 19 point cell *)

xx[0] = 0; yy[0] = 0;  zz[0] = 0;
xx[1] = 1; yy[1] = 0;  zz[1] = 0;
xx[2] = 0; yy[2] = 1;  zz[2] = 0;
xx[3] =-1; yy[3] = 0;  zz[3] = 0;
xx[4] = 0; yy[4] =-1;  zz[4] = 0;
xx[5] = 1; yy[5] = 0;  zz[5] = 1;
xx[6] = 0; yy[6] = 1;  zz[6] = 1;
xx[7] =-1; yy[7] = 0;  zz[7] = 1;
xx[8] = 0; yy[8] =-1;  zz[8] = 1;
xx[9] = 1; yy[9] = 0;  zz[9] =-1;
xx[10]= 0; yy[10]= 1;  zz[10]=-1;
xx[11]=-1; yy[11]= 0;  zz[11]=-1;
xx[12]= 0; yy[12]=-1;  zz[12]=-1;
xx[13]= 0; yy[13]= 0;  zz[13]= 1;
xx[14]= 0; yy[14]= 0;  zz[14]=-1;
xx[15]= 1; yy[15]= 1;  zz[15]= 0;
xx[16]=-1; yy[16]= 1;  zz[16]= 0;
xx[17]=-1; yy[17]=-1;  zz[17]= 0;
xx[18]= 1; yy[18]=-1;  zz[18]= 0;

    (* Form 19 equations to define the values of u at 19 points *)

Clear[i,j,l];
Clear[eq];
Do[eq[i]=u[i]==p/.{x->xx[i],y->yy[i],z->zz[i]},{i,0,18}]
Clear[var];
var={};
Do[If[i+j+l<=n,var=Append[var,a[i,j,l]],Continue[]],{i,0,n},{j,0,n},{l,0,n}];

     (* Define the differential equation *)
```

```
q=D[p,{x,2}]+D[p,{y,2}]+D[p,{z,2}];

     (* Form 7 more equations by using f and its derivatives *)

eq[19]=b[0,0,0]==(q/.{x->0,y->0,z->0});
eq[20]=b[1,0,0]==(D[q,x]/.{x->0,y->0,z->0});
eq[21]=b[0,1,0]==(D[q,y]/.{x->0,y->0,z->0});
eq[22]=b[0,0,1]==(D[q,z]/.{x->0,y->0,z->0});
eq[23]=b[2,0,0]==(D[q,{x,2}]/.{x->0,y->0,z->0});
eq[24]=b[0,2,0]==(D[q,{y,2}]/.{x->0,y->0,z->0});
eq[25]=b[0,0,2]==(D[q,{z,2}]/.{x->0,y->0,z->0});

     (* Solve the chosen system for a[0,0,0], a[1,0,0], a[0,1,0], a[0,0,1] *)

Eliminate[{eq[1],eq[2],eq[3],eq[4],eq[5],eq[6],eq[7],eq[8],eq[9],
eq[10],eq[11],eq[12],eq[13],eq[14],eq[15],eq[16],eq[17],eq[18],
eq[19],eq[20],eq[21],eq[22],eq[23],eq[24],eq[25]},
{a[2,0,0],a[0,2,0],a[0,0,2],a[1,1,0],a[1,0,1],a[0,1,1],
a[3,0,0],a[0,3,0],a[0,0,3],a[2,1,0],a[1,2,0],a[1,0,2],a[2,0,1],
a[0,1,2],a[0,2,1],a[1,1,1],a[4,0,0],a[0,4,0],a[0,0,4],
a[2,2,0],a[2,0,2],a[0,2,2],a[3,1,0],a[1,3,0],a[0,3,1],a[0,1,3],
a[1,0,3],a[3,0,1],a[1,1,2],a[2,1,1],a[1,2,1]}]

b[0, 0, 0] == -4*a[0, 0, 0] - 2*a[0, 0, 6] - (2*a[0, 2, 4])/3 -
(2*a[0, 4, 2])/3 - 2*a[0, 6, 0] - (2*a[2, 0, 4])/3 - (2*a[2, 4, 0])/3 -
(2*a[4, 0, 2])/3 - (2*a[4, 2, 0])/3 - 2*a[6, 0, 0] - b[0, 0, 2]/12 -
b[0, 2, 0]/12 - b[2, 0, 0]/12 + u[1]/3 + u[2]/3 + u[3]/3 + u[4]/3 +
u[5]/6 + u[6]/6 + u[7]/6 + u[8]/6 + u[9]/6 + u[10]/6 + u[11]/6 +
u[12]/6 + u[13]/3 + u[14]/3 + u[15]/6 + u[16]/6 + u[17]/6 + u[18]/6 &&

b[0, 0, 1] == -6*a[0, 0, 1] - 6*a[0, 0, 5] - 2*a[0, 2, 3] -
2*a[0, 4, 1] - 2*a[2, 0, 3] - 2*a[4, 0, 1] + u[5]/2 + u[6]/2 + u[7]/2 +
u[8]/2 - u[9]/2 - u[10]/2 - u[11]/2 - u[12]/2 + u[13] - u[14] &&

b[0, 1, 0] == -6*a[0, 1, 0] - 2*a[0, 1, 4] - 2*a[0, 3, 2] -
6*a[0, 5, 0] - 2*a[2, 3, 0] - 2*a[4, 1, 0] + u[2] - u[4] + u[6]/2 -
u[8]/2 + u[10]/2 - u[12]/2 + u[15]/2 + u[16]/2 - u[17]/2 - u[18]/2 &&

b[1, 0, 0] == -6*a[1, 0, 0] - 2*a[1, 0, 4] - 2*a[1, 4, 0] - 2*a[3, 0, 2] -
2*a[3, 2, 0] - 6*a[5, 0, 0] + u[1] - u[3] + u[5]/2 - u[7]/2 + u[9]/2 -
u[11]/2 + u[15]/2 - u[16]/2 - u[17]/2 + u[18]/2

Solve[%,{a[0,0,0],a[1,0,0],a[0,1,0],a[0,0,1]}]

a[0, 0, 0] -> (-24*a[0, 0, 6] - 8*a[0, 2, 4] - 8*a[0, 4, 2] -
24*a[0, 6, 0] - 8*a[2, 0, 4] - 8*a[2, 4, 0] - 8*a[4, 0, 2] - 8*a[4, 2, 0] -
24*a[6, 0, 0] - 12*b[0, 0, 0] - b[0, 0, 2] - b[0, 2, 0] - b[2, 0, 0] +
4*u[1] + 4*u[2] + 4*u[3] + 4*u[4] + 2*u[5] + 2*u[6] + 2*u[7] + 2*u[8] +
2*u[9] + 2*u[10] + 2*u[11] + 2*u[12] + 4*u[13] + 4*u[14] + 2*u[15] +
```

```
2*u[16] + 2*u[17] + 2*u[18])/48

a[1, 0, 0] -> (-4*a[1, 0, 4] - 4*a[1, 4, 0] - 4*a[3, 0, 2] - 4*a[3, 2, 0] -
12*a[5, 0, 0] - 2*b[1, 0, 0] + 2*u[1] - 2*u[3] + u[5] - u[7] + u[9] -
u[11] + u[15] - u[16] - u[17] + u[18])/12

a[0, 1, 0] -> (-4*a[0, 1, 4] - 4*a[0, 3, 2] - 12*a[0, 5, 0] - 4*a[2, 3, 0] -
4*a[4, 1, 0] - 2*b[0, 1, 0] + 2*u[2] - 2*u[4] + u[6] - u[8] + u[10] -
u[12] + u[15] + u[16] - u[17] - u[18])/12

a[0, 0, 1] -> (-12*a[0, 0, 5] - 4*a[0, 2, 3] - 4*a[0, 4, 1] - 4*a[2, 0, 3] -
4*a[4, 0, 1] - 2*b[0, 0, 1] + u[5] + u[6] + u[7] + u[8] - u[9] - u[10] -
 u[11] - u[12] + 2*u[13] - 2*u[14])/12
```

The output equation for $a[0,0,0]$ contains sixth degree terms such as $a[6,0,0]$ representing $h^6 u_{xxxxxx}$. The output equations for $a[1,0,0]$ (representing hu_x) contain fifth degree terms such as $a[5,0,0]$ representing $h^5 u_{xxxxx}$. All these equations represent finite difference approximations of order h^4.

The four output equations provide the main finite difference scheme and the approximations of fourth order for the first derivatives of the unknown:

$$2\Diamond u_0 + \Box u_0 - 24u_0 = 6b_{0,0,0}h^2 + (b_{2,0,0} + b_{0,2,0} + b_{0,0,2})h^4$$

$$\frac{\partial u}{\partial x} = \frac{1}{12h}(2u_1 - 2u_3 + u_5 - u_7 + u_9 - u_{11} + u_{15} - u_{16} - u_{17} + u_{18} - 2h^3\frac{\partial f}{\partial x})$$

$$\frac{\partial u}{\partial y} = \frac{1}{12h}(2u_2 - 2u_4 + u_6 - u_8 + u_{10} - u_{12} + u_{15} + u_{16} - u_{17} - u_{18} - 2h^3\frac{\partial f}{\partial y})$$

$$\frac{\partial u}{\partial z} = \frac{1}{12h}(2u_{13} - 2u_{14} + u_5 + u_6 + u_7 + u_8 - u_9 - u_{10} - u_{11} - u_{12} - 2h^3\frac{\partial f}{\partial z})$$

This 19-point formula is related to the Mehrstellen scheme for 2D Poisson equation [12] and had also been obtained by other authors. We showed that this scheme is stable and achieves fourth-order accuracy. The finite difference approximation for the gradients of the solution u were presented in this work by us for the first time in the literature. Similar derivative gradient approximations were obtained by us [22] for the 2D Poisson equation and their utilization with 2D Navier–Stokes equations was carried out in [23].We

showed in [29] that these approximations indeed provided solutions of fourth-order accuracy.

3. Biharmonic equation: use of unknown and its first derivatives

Biharmonic equation has been of particular interest because it represents the Stokes' flow and also represents the linearized form of the Navier–Stokes equations. We worked on both two- and three-dimensional problems:

$$\triangle^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy} = f \qquad (2D)$$
$$\triangle^2 u = u_{xxxx} + u_{yyyy} + u_{zzzz} + 2u_{xxyy} + 2u_{yyzz} + 2u_{zzxx} = f \qquad (3D)$$

These are fourth order differential equations and the conventional finite difference approximations for solving the biharmonic equation can be summarized as follows:

1. 13 point approximations in 2-dimensions: truncation error of order h^2
2. 25 point approximations in 2-dimensions: truncation error of order h^4
3. 25 point approximations in 3-dimensions: truncation error of order h^2

The coupled equation approach has been very popular over the past five decades. It involves splitting the biharmonic equation into two Poisson equations which can then be solved using any of the available Poisson solvers:

$$\triangle^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy} = f$$

which is split into two Poisson Equations

$$\triangle u(x, y) = u_{xx} + u_{yy} = v(x, y)$$
$$\triangle v(x, y) = v_{xx} + v_{yy} = f(x, y)$$

There have been difficulties with both of the conventional approaches:

1. The 13 point formula for 2D case requires special treatment near the boundaries because $u(x_i, y_j)$ is connected to neighbouring values two grid points away in each direction. This requires special finite difference approximations for grid points close to the boundaries. The situation is similar in 3D case, and is worse with the $O(h^4)$ conventional approximations [33].
2. Large computational resources are required because direct solvers can only be used for moderate values of grid width h and the conventional iterative methods either converge very slowly or diverge [23, 32].

3. The coupled equation approach requires the boundary values of $v(x, y)$ that are usually unknown and need to be approximated from $v(x, y) = u_{xx} + u_{yy}$ [13, 17].

We proposed a compact solution approach where we discretized the biharmonic equation using the values of the unknown solution u, and the values of gradients u_x, u_y (& u_z for 3D) at selected grid points. The reasons for using the grid values of these gradients are:

1. When the grid values of the gradients u_x, u_y & u_z are needed, they do not need to be approximated using further approximations.
2. Given boundary conditions of the first kind (u, u_n) are exactly satisfied and no special approximations are needed near the boundaries.
3. Proposed finite difference approximations are derived on a compact cell and no special modifications of the finite difference approximations are needed close to the boundaries.

Compact finite difference approximation of the two-dimensional biharmonic equation using 9 grid points is given below. This approximation has a truncation error of $O(h^2)$ [1].

$$\begin{aligned} &28u_{i,j} - 8(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1}) \\ &\quad + (u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i+1,j-1}) \\ &\quad + 3h(u_{x_{i+1,j}} - u_{x_{i-1,j}} + u_{y_{i,j+1}} - u_{y_{i,j-1}}) = \frac{h^4}{2} f_{i,j} \end{aligned}$$

Compatible fourth order approximations for the first derivatives are defined by

$$\begin{aligned} hu_{x_{i,j}} &= (3/4)(u_{i+1,j} - u_{i-1,j}) - (h/4)(u_{x_{i+1,j}} + u_{x_{i-1,j}}) \\ hu_{y_{i,j}} &= (3/4)(u_{i,j+1} - u_{i,j-1}) - (h/4)(u_{y_{i,j+1}} + u_{y_{i,j-1}}) \end{aligned}$$

Compact finite difference approximation of the two-dimensional biharmonic equation with truncation error of order $O(h^4)$ is given below. This approximation also uses 9 grid points [2].

$$\begin{aligned} u_{i,j} &- \frac{3}{11}(u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1}) \\ &+ \frac{1}{44}(u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i+1,j-1}) \\ &+ \frac{7h}{66}(u_{x_{i+1,j}} - u_{x_{i-1,j}} + u_{y_{i,j+1}} - u_{y_{i,j-1}}) \end{aligned}$$

$$+\frac{h}{264}(u_{x_{i+1,j+1}}-u_{x_{i-1,j+1}}-u_{x_{i-1,j-1}}+u_{x_{i+1,j-1}}+u_{y_{i+1,j+1}}$$
$$+u_{y_{i-1,j+1}}-u_{y_{i-1,j-1}}-u_{y_{i+1,j-1}})$$
$$=\frac{h^4}{396}[7f_{i,j}+2(f_{i+1,j}+f_{i,j+1}+f_{i-1,j}+f_{i,j-1})]$$

Compact finite difference approximation for three-dimensional biharmonic equation using 19 grid points in a unit cube is given below. This approximation has a truncation error of $O(h^2)$ [2].

$$\begin{aligned}
&48u_{i,j,k}-10(u_{i+1,j,k}+u_{i,j+1,k}+u_{i,j,k+1}+u_{i-1,j,k}+u_{i,j-1,k}+u_{i,j,k-1})\\
&+(u_{i+1,j,k+1}+u_{i,j+1,k+1}+u_{i-1,j,k+1}+u_{i,j-1,k+1}+u_{i+1,j,k-1}+u_{i,j+1,k-1}\\
&+u_{i-1,j,k-1}+u_{i,j-1,k-1}+u_{i+1,j+1,k}+u_{i-1,j+1,k}+u_{i-1,j-1,k}+u_{i+1,j-1,k})\\
&+3h(u_{x_{i+1,j,k}}-u_{x_{i-1,j,k}}+u_{y_{i,j+1,k}}-u_{y_{i,j-1,k}}+u_{z_{i,j,k+1}}-u_{z_{i,j,k-1}})\\
&=(h^4/2)f_{i,j,k}
\end{aligned}$$

Compatible fourth order approximations for the first derivatives in 3D case are given by:

$$\begin{aligned}
hu_{x_{i,j,k}}&=(3/4)(u_{i+1,j,k}-u_{i-1,j,k})-(h/4)(u_{x_{i+1,j,k}}+u_{x_{i-1,j,k}})\\
hu_{y_{i,j,k}}&=(3/4)(u_{i,j+1,k}-u_{i,j-1,k})-(h/4)(u_{y_{i,j+1,k}}+u_{y_{i,j-1,k}})\\
hu_{z_{i,j,k}}&=(3/4)(u_{i,j,k+1}-u_{i,j,k+1})-(h/4)(u_{y_{i,j,k+1}}+u_{y_{i,j,k-1}})
\end{aligned}$$

Here are the sample results of numerical experiments on the biharmonic equation using our compact formulations. Further examples are available in the cited references.

For a two-dimensional test problem, we considered

$$\triangle^2 u=u_{xxxx}+u_{yyyy}+2u_{xxyy}=f$$

on a unit square with exact solution $u(x,y)=x^3log(1+y)+y/(1+x)$. The forcing term f and the boundary values of u and its derivatives are obtained from the exact solution [1]. An $O(h^4)$ rate of convergence was observed.

Grid	Error	No. of W-cycles
16^2	8.9×10^{-8}	17
32^2	5.8×10^{-9}	17
64^2	4.1×10^{-10}	17

We note that the errors exhibited in this table decrease by a factor of 16 when the grid size is halved; this clearly represents a $O(h^4)$ rate of

convergence. As a comparison, Linden [1985] obtained the maximum error of 2.02×10^{-7} with a 256×256 grid while our method produced 8.9×10^{-8} error with a 16×16 grid.

For a three-dimensional test problem, we considered the biharmonic equation on a unit cube [2]

$$\triangle^2 u = u_{xxxx} + u_{yyyy} + u_{zzzz} + 2u_{xxyy} + 2u_{yyzz} + 2u_{zzxx} = f$$

with exact solution

$$u(x, y, z) = (1 - cos(2\pi x))(1 - cos(2\pi y))(1 - cos(2\pi z)).$$

The forcing term f and boundary values of u and u_n were obtained from exact solution u. The results are summarized in the following table:

Grid	Error	time in secs	No. of W-cycles
16^3	3.9×10^{-5}	64.7	20
32^3	2.3×10^{-6}	679.5	22
64^3	3.0×10^{-7}	5747.9	22

We note that the results for 3D problem were obtained using multigrid technology for grid sizes as small as $64 \times 64 \times 64$ which meant solving linear systems with $274,625$ unknowns. I first announced these results in March 2001 at the Second International Seminar on Numerical Analysis in Engineering (NAE2001) in Batam Island, Indonesia. Results for the two-dimensional biharmonic equation were published in SIAM Journal on Scientific Computing (1998) [1]. The results for 3D biharmonic equation were published in Numerical Algorithms (2002) [2]. It is noted that we used Mathematica to automate the derivation of finite difference approximations for biharmonic equations; these Mathematica codes are available in [1].

4. Navier–Stokes equations: steady state

The Navier–Stokes (N–S) equations in primitive variable formulation may be written as following:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\nabla^2 u$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\nabla^2 v$$

Here u, v are the velocities along x- and y-directions respectively, p the pressure, t the time, $Re = \frac{U_0 L}{\nu}$ is the Reynolds number with U_0 and L being some characteristic velocity and length, and ν the kinematic viscosity.

Introducing streamfunction ψ and vorticity ω as $u = \frac{\partial \psi}{\partial y}$, $v = -\frac{\partial \psi}{\partial x}$, $\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$, the above N–S equations can be rewritten as:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega$$

$$\frac{\partial \omega}{\partial t} = \frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) - \left(u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y}\right)$$

This is known as Streamfunction-Vorticity (ψ-ω) formulation which has been used in many decades of computations for the resolution of fluid flows. This formulation has been very successful and has been used by a large number of researchers over the past many decades to test new methods for the numerical solutions of a variety of fluid flow problems. Typical difficulty with this formulation consists in the specification of vorticity values at the no-slip boundaries. The vorticity ω is defined through the Poisson equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega$$

which needs to be solved discretely on the boundaries so that boundary values of the vorticity can be specified for the vorticity transport equation when this formulation is utilized. As the values of vorticity ω on the boundaries are generally unspecified, one has to carry out some kind of numerical approximations in order to specify the boundary values of vorticity and thus being able to solve the above equations.

Inspired by our work on the biharmonic equation, we extended it to our eventual target: the Navier–Stokes equations. The derivation of second and fourth approximations of the Navier–Stokes equations was carried out with the help of Mathematica (Code to be presented later). While the fourth order approximations are available, we have found that the second order formulations are adequate for most practical fluid flow computations. Some of these results were announced by the author at the International Conference on Industrial and Applied Mathematics (ICIAM 2003) in Sydney, Australia in 2003.

Our derivation of the streamfunction-velocity formulation of the 2D Navier–Stokes Equations considered the fourth order differential equation

(representing the pure streamfunction formulation of N–S equations):

$$\frac{\partial^4\psi}{\partial x^4} + 2\frac{\partial^4\psi}{\partial x^2\partial y^2} + \frac{\partial^4\psi}{\partial y^4} - Reu\left[\frac{\partial^3\psi}{\partial x^3} + \frac{\partial^3\psi}{\partial x\partial y^2}\right] - Rev\left[\frac{\partial^3\psi}{\partial y^3} + \frac{\partial^3\psi}{\partial x^2\partial y}\right] = 0$$

which can also be written as

$$\nabla^4\psi = Rev\nabla^2 u - Reu\nabla^2 v$$

The second order finite difference scheme for the above differential equation is

$$\begin{aligned} 28\psi_{i,j} &- 8(\psi_{i-1,j} + \psi_{i,j-1} + \psi_{i+1,j} + \psi_{i,j+1}) \\ &+ (\psi_{i-1,j+1} + \psi_{i-1,j-1} + \psi_{i+1,j+1} + \psi_{i+1,j-1}) \\ &= 3h(u_{i,j-1} - u_{i,j+1} + v_{i+1,j} - v_{i-1,j}) \\ &+ Re\frac{h^2}{4}[v_{i,j}(u_{i,j-1} + u_{i,j+1} + u_{i+1,j} + u_{i-1,j}) \\ &- u_{i,j}(v_{i,j-1} + v_{i,j+1} + v_{i+1,j} + v_{i-1,j})]. \end{aligned}$$

We announced the streamfunction-velocity formulation for the Navier–Stokes equations in 2005 ([25], Journal of Computational Physics). This method was used to solve the following problems

1. Driven cavity problem
2. Two-sided lid driven cavity Problem
3. Backward step problem

Here we give the Mathematica code for deriving the streamfunction-velocity formulation for the fourth order differential equation representing Navier–Stokes equations. This code has not appeared in any of our previous publications. Similar code for the fourth order formulation is also available.

```
Navier Stokes Equations second order Formulation using Mathematica

Clear[p,q,r,i,j,k,xx,yy,u,ux,uy,eq,equ,a,b]; n=6;
p=0;Clear[i,j,a]
Do[If[i+j<=n,p=p+a[i,j] x^i y^j,Continue[]],{i,0,n},{j,0,n}];
   dpx=D[p,x];dpy=D[p,y];
xx[0]=0;yy[0]=0;xx[1]=1;yy[1]=0;
xx[2]=0;yy[2]=1;xx[3]=-1;yy[3]=0;
xx[4]=0;yy[4]=-1;xx[5]=1;yy[5]=1;
xx[6]=-1;yy[6]=1;xx[7]=-1;yy[7]=-1;xx[8]=1;yy[8]=-1;
```

```
Clear[i,j];Clear[eq];
Do[eq[i]=u[i]==p/.{x->xx[i],y->yy[i]},{i,0,8}]
Do[eq[i+9]=ux[i]==dpx/.{x->xx[i],y->yy[i]},{i,0,8}]
Do[eq[i+18]=uy[i]==dpy/.{x->xx[i],y->yy[i]},{i,0,8}] Clear[ c1, c2]

q=D[p,{x,4}]+2 D[p,{x,2},{y,2}]+ D[p,{y,4}]+c1 (D[p,{x,2},y]+D[p,{y,3}])
   +c2 (D[p,{x,3}] + D[p,x,{y,2}])

eq[27]=b[0,0]==(q/.{x->0,y->0});
eq[28]=b[1,0]==(D[q,x]/.{x->0,y->0});
eq[29]=b[0,1]==(D[q,y]/.{x->0,y->0});
eq[30]=b[2,0]==(D[q,{x,2}]/.{x->0,y->0});
eq[31]=b[1,1]==(D[q,x,y]/.{x->0,y->0});
eq[32]=b[0,2]==(D[q,{y,2}]/.{x->0,y->0});

Eliminate[{eq[1],eq[2],eq[3],eq[4],eq[5],eq[6],eq[7],eq[8], eq[10],eq[20],
   eq[12],eq[22],
(*eq[14],eq[15],eq[16],eq[17],
eq[23],eq[24],eq[25],eq[26],*) eq[27],eq[28],eq[29],eq[30](*,eq[31]*),
   eq[32]},
{a[2,0],a[1,1],a[0,2],
a[3,0],a[2,1],a[1,2],a[0,3],
a[4,0],a[3,1],a[2,2],a[1,3],a[0,4](*,
a[5,0],a[4,1],a[3,2],a[2,3],a[1,4],a[0,5],
a[6,0],a[5,1],a[4,2],a[3,3],a[2,4],a[1,5],a[0,6]*)}]

Solve[%,{a[0,0],a[1,0],a[0,1]}]

Simplify[%]
```

The first test problem for our new formulation is the two-dimensional lid-driven square cavity flow (see the schematic in Figure 5) which is extensively used as a benchmark for code validation of the incompressible N–S equations. The cavity is defined in the unit square and the fluid motion is generated by the sliding motion of the top wall of the cavity ($y = 1$) in its own plane from left to right. Boundary conditions on the top wall are given as $u = 1, v = 0$. On all other walls of the cavity the velocities are zero ($u = v = 0$). Further, the streamfunction values on all four walls are zero ($\psi = 0$).

The solutions of this problem are well documented and details can be found in the cited references. We simply exhibit the streamline contours for some Reynolds numbers (Figure 6) from [25]:

Next we considered the flow over the backward-facing step in a channel which provides an excellent test case for the accuracy of numerical methods because the reattachment length is a function of the Reynolds number. The problem configuration is shown in Figure 7 [26]. Numerical simulations were

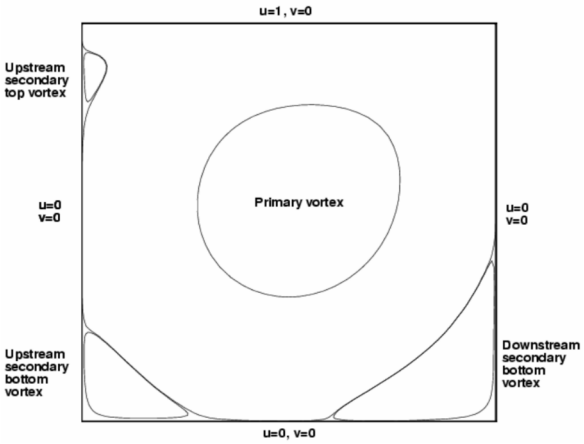

Figure 5: The schematic of lid driven square cavity flow.

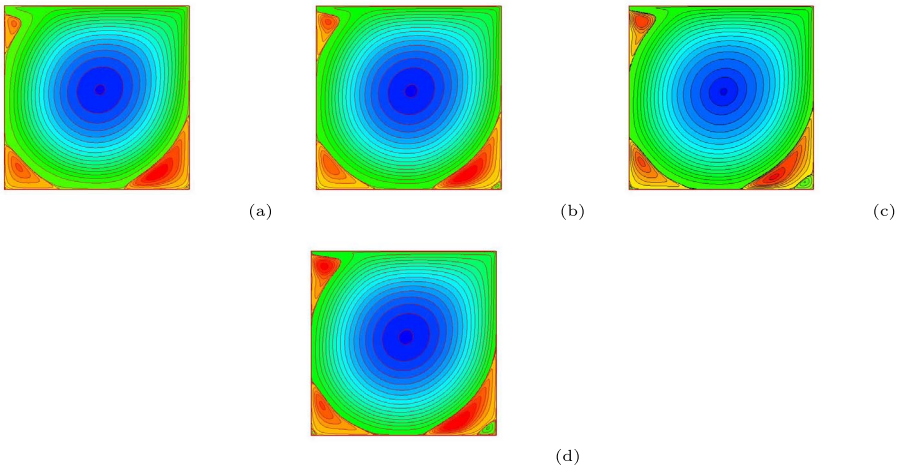

Figure 6: Streamlines contours (a) $Re = 3200$, (b) $Re = 5000$, (c) $Re = 7500$ and (d) $Re = 10000$.

carried out for the range of Reynolds numbers (Re) from 100 to 1000 on grid sizes ranging from 601×21 to 961×33. At the inlet, a parabolic velocity profile is usually prescribed and we used $u = 24y(0.5 - y), v = 0$. The downstream boundary conditions were prescribed at a distance of 30 step heights so as to allow the flow to be fully developed. Thus at the outlet, $\partial u/\partial x = 0$ and $v = 0$. At the stationary walls, $u = v = 0$ and $\psi = 0.5$. At

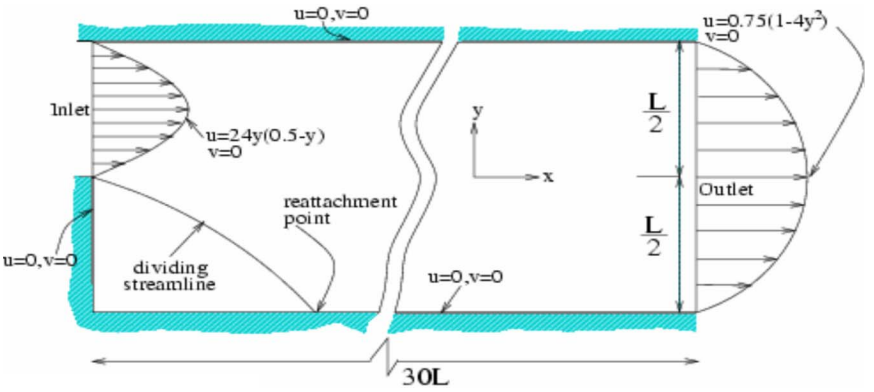

Figure 7: Schematic of backward facing step flow, [26].

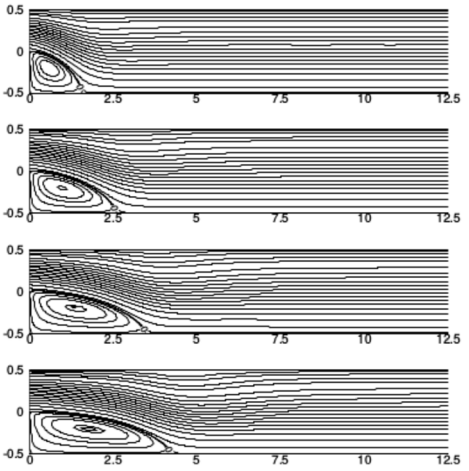

Figure 8: For Re = 100, 200, 300, 400 from top to bottom, [26].

the inlet, the value of ψ can be found from the prescribed values of u and v and at the outlet the Neumann boundary condition $\partial\psi/\partial x = 0$ is used.

We exhibit (in Figure 8) a few results obtained with a fine 961×33 grid that was used to accurately capture the recirculation zone in the vicinity of the step, and the upper wall eddy when the Reynolds number is large. In these figures, one can see a steady increase in the reattachment length with the increase in Re. The formation of a secondary vortex at the upper wall was also seen for higher values of Reynolds number [26]. Figure 9 exhibits the velocity profiles at the outlet.

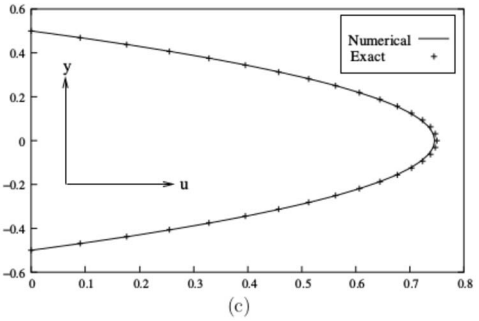

Figure 9: Exact (obtained from analytical expression) and numerical streamwise velocity profile at the outlet.

A team of researchers including Ben–Artzi and Fishelov have also been working on the streamfunction formulation of Navier–Stokes equations and have published several articles in this area [6, 4, 5, 14]. Another team consisting of Tian and Yu have also published results in this area [62, 63, 64, 65].

5. Navier–Stokes equations: time dependent case

Once we were successful with the steady state form of the Navier–Stokes equations, we extended the $\psi - v$ approach to unsteady N–S equations and proposed a second-order implicit, unconditionally stable $\psi - v$ formulation for the unsteady incompressible N–S equations. The method was used to solve several 2D time dependent fluid flow problems, including the flow decayed by viscosity problem (with analytical solution), the lid-driven square cavity problem, the backward-facing step problem, and the flow past a square prism problem. For the problems with known exact solutions, our coarse grid transient solutions were extremely close to the analytical solutions even for high Reynolds numbers (Re). For the driven cavity problem, our time-marching steady-state solutions up to $Re = 7500$ provided excellent matches with established numerical results, and for $Re = 10000$, we concluded that the asymptotic stable solution is indeed periodic as has also been found by other authors. For the backward-facing step problem, our numerical results were in excellent agreement with established numerical and experimental results. Finally, for the flow past a square prism, we have very successfully simulated the von Karman vortex street for $Re = 200$.

We presented the ψ-v approach of the N–S equation to its transient counterpart in 2010 ([42], International Journal for Numerical Methods in Fluids)

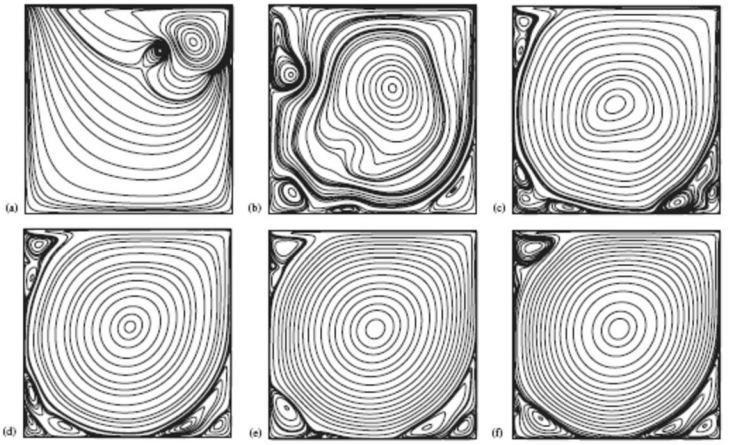

Figure 10: For Re = 10000: t = (a) 5; (b) 25; (c) 50; (d) 100; (e) 900 and (f) 1600.

where we approximated the time derivative through a Crank–Nicolson type discretization. The following problems were considered:

1. Driven Cavity Problem: Time-marching solutions up to $Re = 10{,}000$. Evidence that the asymptotic stable solution is periodic.
2. Backward-facing step problem.
3. Flow past a square prism exhibiting von Karman vortex street at moderate Reynolds numbers [42].

Here are some of these results:

Evolution of streamfunction for the lid-driven square cavity flow [42]: In Figure 10, we display the evolution of the solutions for $Re = 10000$ from $t = 5$ to $t = 1600$ starting with a zero initial profile. This set of profiles shows that the steady state is unstable for this value of Reynolds number. Some recent studies have concluded that a stable steady-state solution for this problem is not possible beyond a critical Reynolds number which is close to $Re = 8000$. Our computations for $Re = 10000$ support such a conclusion and lean towards an almost periodic solution.

Periodic flow for the lid-driven square cavity flow for Re = 10000 is demonstrated in Figure 11 [42]. This figure contains a sequence of six streamline contours during one main period from time $t = T_0$ to $t = T_0 + 2.345$ with $T_0 = 1500$ such that the six plots make one complete cycle. We observe that there are persistent oscillations in all of the secondary and tertiary vortices,

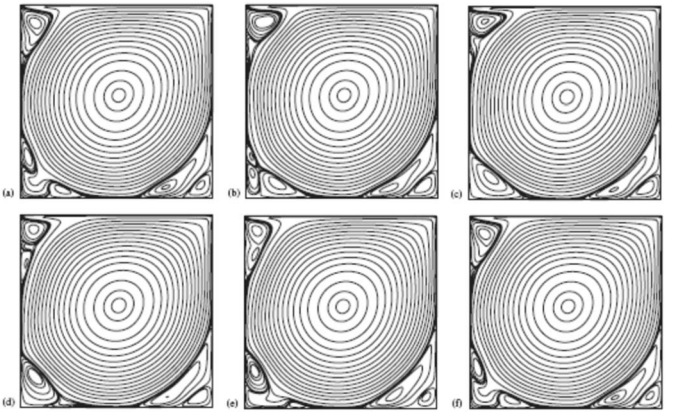

Figure 11: Evolution of the streamfunction and cyclic nature of the flow for one main period with $t = T_0 + \alpha$ where $T_0 = 1500$ and (a) $\alpha = 0.0$, (b) $\alpha = 0.469$, (c), $\alpha = 0.938$, (d) $\alpha = 1.407$, (e) $\alpha = 1.876$ and (f) $\alpha = 2.345$.

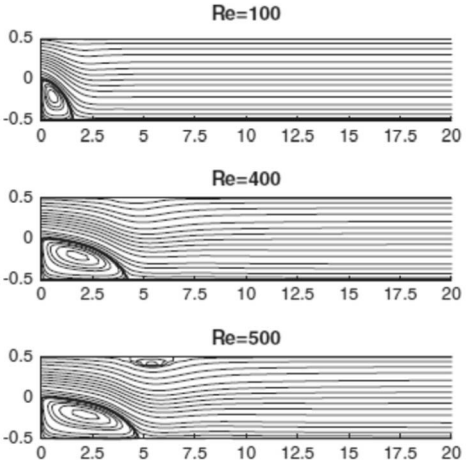

Figure 12: For Re = 100, 400, 500.

particularly on the top left and bottom left walls while the secondary and tertiary vortices at the bottom right wall are almost stable.

We now consider the flow over the backward-facing step in a channel; this example provides an excellent test case for the accuracy of our numerical method because the reattachment length is a function of the Reynolds number. Numerical simulations were carried out for Reynolds numbers $100 - 800$ on grid sizes ranging from 601×21 to 961×33. Figures 12 and 13 present

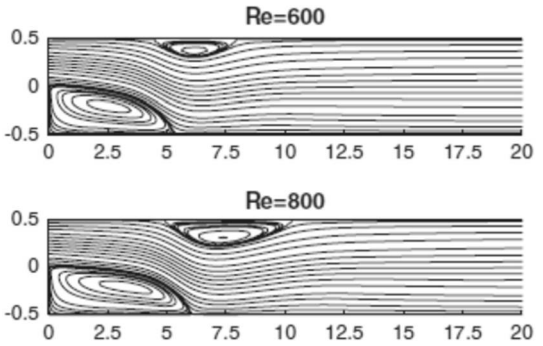

Figure 13: For Re = 600 and 800.

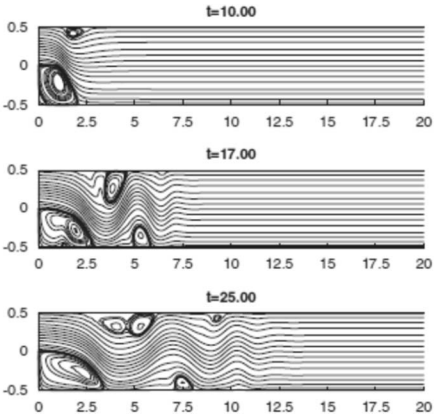

Figure 14: For Re = 600 at time stations t = 10, 17, 25 s.

the steady-state streamlines for the backward-facing step flow [42]. These figures show the steady-state streamline patterns for $100 \leq Re \leq 800$. One can see a steady increase in the reattachment length with the increase in Re. The formation of a secondary vortex at the upper wall can be seen from $Re = 500$ onwards. In Figures 14 and 15, we show the time evolution of the streamlines for $Re = 600$. One can see the development of the main vortices at the step and on the top wall; a secondary vortex appears at the bottom left corner starting at $t = 10$. After a while (as seen at $t = 17$ and $t = 25$), two smaller vortices begin appearing at the top and bottom walls; these vortices diffuse completely before the steady state is reached. The main vortex at the top wall splits into two vortices (see Figure 14 at $t = 25$) and within a short period of time these vortices coalesce into one vortex (see Figure 15 at $t = 29$). Both of the main vortices grow in size with time. Once the

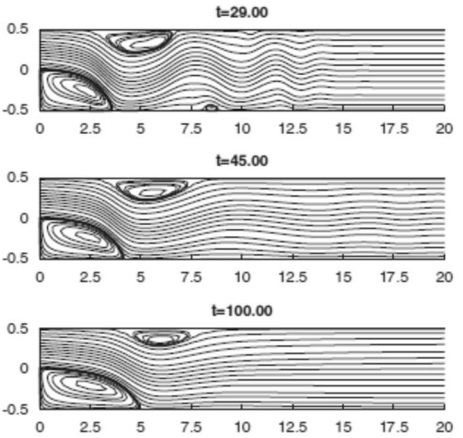

Figure 15: For Re = 600 at time stations t = 29, 45 and 100 s.

steady state is reached, only the vortices at the top wall, at the step and the secondary vortex at the bottom left corner (not visible in the figures) continue to exist.

As our final test problem for time-dependent N–S equations, we solved the problem of flow past a square prism. The domain of the flow is multiply connected and the nature of the flow itself is very complex. The unsteady behaviour of the flow that evolves with time for Reynolds numbers beyond a critical value makes this problem more challenging and interesting. Thus this problem serves as a suitable test case to check the robustness and reliability of our numerical scheme.

In Figure 16, we give the configuration of the flow past a square cylinder problem [42]. As seen in Figure 17, a symmetric flow was observed at the beginning, but the flow became unstable later on, and finally the flow lost its symmetry. Eventually, the flow settled into a periodic nature (figure (f) for $t = 360$). Figure 17 also exhibits the evolution of streamfunction towards a periodic state for the flow past a square prism problem. In Figures 18 and 19, we present the temporal evolution of streamlines and vorticity over one complete vortex shedding cycle of duration T. The evolution of an impressive von Karman vortex street, which is a regular feature of the Reynolds number considered here, is clearly seen in these figures.

My colleague, Professor Jiten C Kalita, with his students and co-authors has been tirelessly working on higher order compact schemes on nonuniform grids [40, 41, 50] and has further extended the compact formulations to non-rectangular domains for steady and transient flow problems [40, 41, 50, 47, 53, 55, 54, 56, 52, 51].

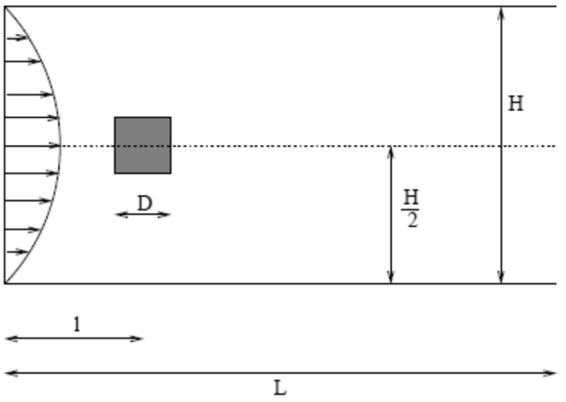

Figure 16: Configuration of the flow past a square cylinder problem.

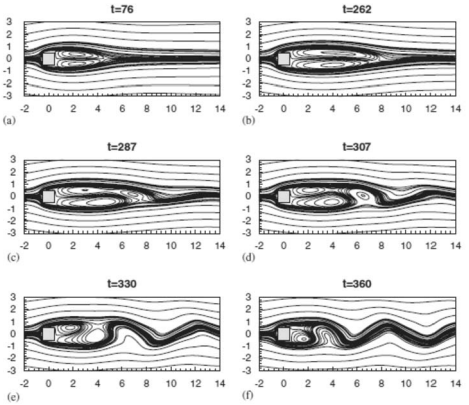

Figure 17: Streamfunction evolution towards a periodic state for Re = 200: (a) t = 76; (b) t = 262; (c) t = 287; (d) t = 307; (e) t = 330; and (f) t = 360s, [42].

6. Other applications

We recently presented an optimization strategy for implementing the BiCGStab iterative solver on graphic processing units (GPU) for computing incompressible viscous flows governed by the unsteady Navier–Stokes *(N–S)* equations on a CUDA platform. Our $\psi - v$ formulation was used to discretize the N–S equations and we obtained remarkable speedup of 40 times on finer grids for the lid-driven square cavity flow. The GPU implementation enabled us to compute the flow for extremely fine grids (1024×1024) and we

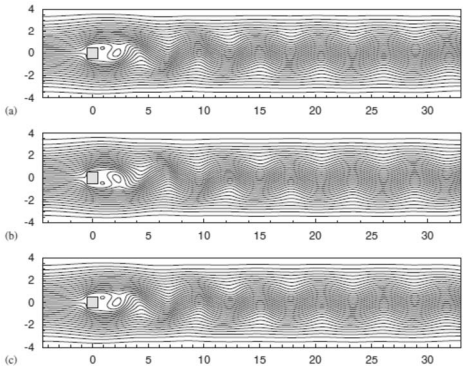

Figure 18: Streamfunction contours depicting the wake behind three successive instants of time over one vortex shedding period: (a) $t = t_0$; (b) $t = t_0 + T/2$ and (c) $t = t_0 + T$.

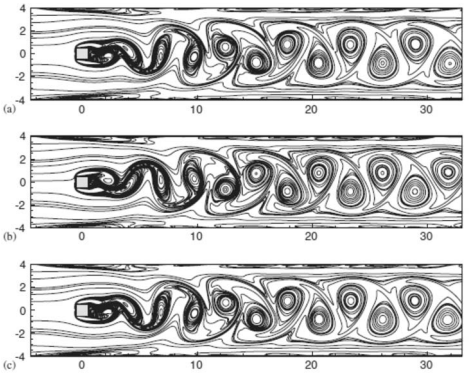

Figure 19: Vorticity contours depicting the wake behind three successive instants of time over one vortex shedding period: (a) $t = t_0$; (b) $t = t_0 + T/2$ and (c) $t = t_0 + T$.

were able to resolve very small scales with remarkable accuracy. While no perceptible gain in speed could be seen on coarser grids, we observed that the speed-up increases with the increase in the problem size. It is heartening to note that on the finest grid, viz., 1024×1024 grid, the speedup is around 40. Successful implementation of the optimization strategy of the BiCGStab algorithm on GPU clearly indicates that it has more potential for efficient computation of even more complex flows.

These results are available in the following articles [44, 45].

7. Conclusions

As shown in this article, we have come a long way from the pencil-and-paper environment when solving a 40×40 problem was a big thing, to being able to solve the two- and three-dimensional problems using 1024×1024 and $64 \times 64 \times 64$ grids. Computer algebra systems like Mathematica have also been invaluable as is the availability of high speed computational architectures.

1. We have derived innovative, compact, formulations for partial differential equations, including convection-diffusion equations, biharmonic equation in two and three dimensions, and Navier–Stokes equations (steady state as well as time-dependent).
2. $O(h^2)$ and $O(h^4)$ approximations are available.
3. Finite difference approximations are derived on compact stencils using values of the solution and its gradients as the unknowns.
4. For the equations governing incompressible viscous flows, this corresponds to the streamfunction-velocity $(\psi - v)$ formulation.
5. Our method is very efficient and allows us to obtain high accuracy solutions for a variety of partial differential equations in a computationally efficient procedure.

Acknowledgement

I would like to thank Professor Tony Wen–Hann Sheu for inviting me to this wonderful conference in March 2019 at National Taiwan University. I am grateful to my colleague and friend, Professor Jiten C Kalita of Indian Institute of Technology Guwahati for his tremendous help in organizing my presentation in Taiwan and also his help in typesetting this paper. I am also grateful to Mr. Raghav Singhal of Indian Institute of Technology Guwahati who helped tremendously with the typesetting and organizing of my Taiwan presentation and this paper. This work has been generously supported, in part, by the Dean of Columbian College of Arts and Sciences at The George Washington University.

References

[1] Irfan Altas, Jonathan Dym, Murli M Gupta, and Ram P Manohar. Multigrid solution of automatically generated high-order discretizations for the biharmonic equation. *SIAM Journal on Scientific Computing*, 19(5):1575–1585, 1998. MR1618733

[2] Irfan Altas, Jocelyne Erhel, and Murli M Gupta. High accuracy solution of three-dimensional biharmonic equations. *Numerical Algorithms*, 29(1–3):1–19, 2002. MR1896942

[3] Irfan Altas and Murli M Gupta. Iterative methods for fixed point problems on high performance computing environment. In *Proceedings of Complex Systems 1996, Albury, Australia, Complex Systems: From Local Interactions to Global Phenomena*, pages 121–128, 1996. MR1657640

[4] M Ben-Artzi, I Chorev, J-P Croisille, and D Fishelov. A compact difference scheme for the biharmonic equation in planar irregular domains. *Siam J. Numer. Anal.*, 47(4):3087–3108. MR2551159

[5] M Ben-Artzi, J-P Croisille, and D Fishelov. A high order compact scheme for the pure-streamfunction formulation of the Navier–Stokes equations. *J Sci Comput.*, 42:216–250. MR2578035

[6] Matania Ben-Artzi, Jean-Pierre Croisille, Dalia Fishelov, and Shlomo Trachtenberg. A pure-compact scheme for the streamfunction formulation of Navier–Stokes equations. *Journal of Computational Physics*, 205:640–664. MR2134996

[7] G Berikelashvili. The difference schemes of high order accuracy for elliptic equations with lower derivatives. In *Proc. A. Razmadze Math. Inst*, volume 117, pages 1–6, 1998. MR1672926

[8] Givi Berikelashvili, Murli M Gupta, and Bidzina Midodashvili. Method of refinement by higher order differences for 3D Poisson equation with nonlocal boundary conditions. *Mathematical Sciences Letters*, 7(2):71–77, 2018.

[9] Givi Berikelashvili, Murli M Gupta, and Bidzina Midodashvili. On the improvement of convergence rate of difference schemes with high order differences for a convection-diffusion equation. In *AIP Conference Proceedings*, volume 1648, page 470002. AIP Publishing, 2015.

[10] Givi Berikelashvili, Murli M Gupta, and Manana Mirianashvili. Convergence of fourth order compact difference schemes for three-dimensional convection-diffusion equations. *SIAM Journal on Numerical Analysis*, 45(1):443–455, 2007. MR2285863

[11] Roger Bouard and Madeleine Coutanceau. The early stage of development of the wake behind an impulsively started cylinder for $40 < Re < 10^4$. *Journal of Fluid Mechanics*, 101(3):583–607, 1980.

[12] L Collatz. *Numerische Behandlung von Differentialgleichungen.* Springer-Verlag, 1951. MR0043563

[13] Louis W Ehrlich and Murli M Gupta. Some difference schemes for the biharmonic equation. *SIAM Journal on Numerical Analysis*, 12(5):773–790, 1975. MR0408275

[14] D Fishelov, M Ben-Artzi, and J-P Croisille. Recent developments in the pure streamfunction formulation of the Navier–Stokes system. *J Sci Comput.*, 45:238–258. MR2679798

[15] David K Gartling. A test problem for outflow boundary conditions flow over a backward-facing step. *International Journal for Numerical Methods in Fluids*, 11(7):953–967, 1990.

[16] Murli M Gupta. Numerical solution of a second biharmonic boundary value problem. *BIT Numerical Mathematics*, 13(2):160–164, 1973. MR0319388

[17] Murli M Gupta. Discretization error estimates for certain splitting procedures for solving first biharmonic boundary value problems. *SIAM Journal on Numerical Analysis*, 12(3):364–377, 1975. MR0403256

[18] Murli M Gupta. Stability of iterative schemes for nonselfadjoint equations. *Aplikace matematiky*, 21(3):173–184, 1976. MR0403209

[19] Murli M Gupta. A comparison of numerical solutions of convective and divergence forms of the Navier–Stokes equations for the driven cavity problem. *Journal of Computational Physics*, 43(2):260–267, 1981. MR0640358

[20] Murli M Gupta. A fourth order finite difference scheme for two-dimensional elliptic equations, volume 1648, pages 147–154. IMACS/ North-Holland Publishing, 1983. MR0751615

[21] Murli M Gupta. A survey of some second-order difference schemes for the steady-state convection-diffusion equation. *International Journal for Numerical Methods in Fluids*, 3(4):319–331, 1983. MR0714086

[22] Murli M Gupta. A fourth-order Poisson solver. *Journal of Computational Physics*, 55(1):166–172, 1984. MR0757429

[23] Murli M Gupta. High accuracy solutions of incompressible Navier–Stokes equations. *Journal of Computational Physics*, 93(2):343–359, 1991. MR1104356

[24] Murli M Gupta. Numerical methods and software (David Kahaner, Cleve Moler, and Stephen Nash). *SIAM Review*, 33(1):144–147, 1991.

[25] Murli M Gupta and Jiten C Kalita. A new paradigm for solving Navier–Stokes equations: streamfunction-velocity formulation. *Journal of Computational Physics*, 207(1):52–68, 2005. MR2143582

[26] Murli M Gupta and Jiten C Kalita. New paradigm continued: Further computations with streamfunction-velocity formulations for solving Navier–Stokes equations. *Communications in Applied Analysis*, 10(4):461–490, 2006. MR2286492

[27] Murli M Gupta and Jules Kouatchou. Symbolic derivation of finite difference approximations for the three-dimensional Poisson equation. *Numerical Methods for Partial Differential Equations: An International Journal*, 14(5):593–606, 1998. MR1639914

[28] Murli M Gupta, Jules Kouatchou, and Jun Zhang. A compact multigrid solver for convection-diffusion equations. *Journal of Computational Physics*, 132(1):123–129, 1997. MR1440337

[29] Murli M Gupta, Jules Kouatchou, and Jun Zhang. Comparison of second-and fourth-order discretizations for multigrid Poisson solvers. *Journal of Computational Physics*, 132(2):226–232, 1997. MR1444997

[30] Murli M Gupta and Ram P Manohar. Numerical solution of a separtated viscous flow problem by a non-uniform finite-difference net. *International Journal for Numerical Methods in Engineering*, 4(2):251–260, 1972. MR0488842

[31] Murli M Gupta and Ram P Manohar. A critique of a second-order upwind scheme for viscous flow problems. *AIAA Journal*, 16(7):759–761, 1978.

[32] Murli M Gupta and Ram P Manohar. Boundary approximations and accuracy in viscous flow computations. *Journal of Computational Physics*, 31(2):265–288, 1979. MR0533686

[33] Murli M Gupta and Ram P Manohar. Direct solution of the biharmonic equation using noncoupled approach. *Journal of Computational Physics*, 33(2):236–248, 1979. MR0549953

[34] Murli M Gupta and Ram P Manohar. On the use of central difference scheme for Navier–Strokes equations. *International Journal for Numerical Methods in Engineering*, 15(4):557–573, 1980. MR0571079

[35] Murli M Gupta, Ram P Manohar, and Ben Noble. Nature of viscous flows near sharp corners. *Computers & Fluids*, 9(4):379–388, 1981.

[36] Murli M Gupta, Ram P Manohar, and John W Stephenson. A fourth order, cost effective and stable finite difference scheme for the convection-diffusion equation. In *Numerical Properties and Methodologies in Heat Transfer*, pages 201–209, 1983. MR0718868

[37] Murli M Gupta, Ram P Manohar, and John W Stephenson. A single cell high order scheme for the convection-diffusion equation with variable coefficients. *International Journal for Numerical Methods in Fluids*, 4(7):641–651, 1984. MR0754894

[38] Murli M Gupta, Ram P Manohar, and John W Stephenson. High-order difference schemes for two-dimensional elliptic equations. *Numerical Methods for Partial Differential Equations*, 1(1):71–80, 1985. MR0868052

[39] Murli M Gupta and Jun Zhang. High accuracy multigrid solution of the 3D convection-diffusion equation. *Applied Mathematics and Computation*, 113(2–3):249–274, 2000. MR1773499

[40] Jiten C Kalita, Anoop K Dass, and D C Dalal. A transformation-free hoc scheme for steady convection-diffusion on non-uniform grids. *Int. J. Numer. Meth. Fluids*, 44:33–53. MR2026734

[41] Jiten C Kalita, Anoop K Dass, and Nimisha Nidhi. An efficient transient Navier–Stokes solver on compact nonuniform space grids. *Journal of Computational and Applied Mathematics*, 214:148–162. MR2391679

[42] Jiten C Kalita and Murli M Gupta. A streamfunction-velocity approach for 2D transient incompressible viscous flows. *International journal for numerical methods in fluids*, 62(3):237–266, 2010. MR2603367

[43] Jiten C Kalita and Shuvam Sen. The biharmonic approach for unsteady flow past an impulsively started circular cylinder. *Communications in Computational Physics*, 12(4):1163–1182, 2012. MR2923852

[44] Jiten C Kalita, Parikshit Upadhyaya, and Murli M Gupta. Gpu accelerated computation of incompressible viscous flows by the streamfunction-velocity (ψ-v) formulation, volume 1648, page 470004. AIP Publishing, 2015.

[45] Jiten C Kalita, Parikshit Upadhyaya, and Murli M Gupta. Optimized bicgstab based gpu accelerated computation of incompressible viscous flows by the ψ-v formulation. *International Journal of Applied and Computational Mathematics*, 3(1):1477–1495, 2017. MR3722379

[46] Jayant Keskar and D A Lyn. Computations of a laminar backward-facing step flow at re = 800 with a spectral domain decomposition method. *International journal for numerical methods in fluids*, 29(4):411–427, 1999.

[47] Pankaj Kumar and Jiten C Kalita. A transformation-free-v formulation of the Navier–Stokes equations on compact nonuniform grids. *Journal of Computational and Applied Mathematics*, 353:292–317. MR3899101

[48] Sanjiva K Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16–42. MR1188088

[49] Ta Phuoc Loc and R Bouard. Numerical solution of the early stage of the unsteady viscous flow around a circular cylinder: a comparison with experimental visualization and measurements. *Journal of Fluid Mechanics*, 160:93–117, 1985.

[50] H V R Mittal, Jiten C Kalita, and Rajendra K Ray. A class of finite difference schemes for interface problems with an hoc approach. *Int. J. Numer. Meth. Fluids*, 82:567–606. MR3565908

[51] Swapan K Pandit, Jiten C Kalita, and D C Dalal. A fourth-order accurate compact scheme for the solution of steady Navier–Stokes equations on non-uniform grids. *Computers and Fluids*, 37(2):121–134. MR2645516

[52] Swapan K Pandit, Jiten C Kalita, and D C Dalal. A transient higher order compact scheme for incompressible viscous flows on geometries beyond rectangular. *Journal of Computational Physics*, 225:1100–1124. MR2346713

[53] Rajendra K Ray and Jiten C Kalita. A transformation-free hoc scheme for incompressible viscous flows on nonuniform polar grids. *Int. J. Numer. Meth. Fluids*, 62:683–708. MR2605013

[54] Shuvam Sen. Fourth order compact schemes for variable coefficient parabolic problems with mixed derivatives. *Computers and Fluids*, 134–135:81–89. MR3507301

[55] Shuvam Sen. A new family of (5,5)cc-4oc schemes applicable for unsteady Navier–Stokes equations. *Journal of Computational Physics*, 251:251–271. MR3094918

[56] Shuvam Sen and Jiten C Kalita. A 4OEC scheme for the biharmonic steady Navier–Stokes equations in non-rectangular domains. *Computer Physics Communications*, 196:113–133. MR3398979

[57] Shuvam Sen, Jiten C Kalita, and Murli M Gupta. A robust implicit compact scheme for two-dimensional unsteady flows with a biharmonic stream function formulation. *Computers & Fluids*, 84:141–163, 2013. MR3096039

[58] W Z Shen and T P Loc. A coupling finite difference/particle method for the resolution of Navier–Stokes equations in velocity-vorticity form. *Aerospace science and technology*, 2:97–109, 1997. MR1455695

[59] W F Spotz and G F Carey. Extension of high-order compact schemes to time dependent problems. *Numer. Meth. PDE*, 17:657–672. MR1859257

[60] W F Spotz and G F Carey. Formulation and experiments with high-order compact schemes for nonuniform grids. *Int. J. Numer. Meth. Heat and Fluid Flow*, 8:288–303.

[61] W F Spotz and G F Carey. High-order compact scheme for the steady streamfunction vorticity equations. *Int. J. Numer. Meth. in Engrg.*, 38:3497–3512. MR1356346

[62] Zhen F Tian and P X Yu. An efficient compact difference scheme for solving the streamfunction formulation of the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 230:6404–6419. MR2818605

[63] P X Yu and Zhen F Tian. Compact computations based on a stream-function-velocity formulation of two-dimensional steady laminar natural convection in a square cavity. *Physical Review E*, 85:036703.

[64] P X Yu and Zhen F Tian. A compact streamfunction-velocity scheme on nonuniform grids for the 2D steady incompressible Navier–Stokes equations. *Computers and Mathematics with Applications*, 66:1192–1212. MR3096454

[65] P X Yu and Zhen F Tian. A high-order compact scheme for the pure streamfunction (vector potential) formulation of the 3D steady incompressible Navier–Stokes equations. *Journal of Computational Physics*, 382:65–85. MR3904309

[66] Jun Zhang, Lixin Ge, and Jules Kouatchou. A two colorable fourth-order compact difference scheme and parallel iterative solution of the 3D convection diffusion equation. *Mathematics and Computers in simulation*, 54(1–3):65–80, 2000. MR1800105

Murli M. Gupta
Department of Mathematics
The George Washington University
Washington DC
USA
E-mail address: mmg@gwu.edu
URL: home.gwu.edu/~mmg

Received August 27, 2019