INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:1031–1037 (DOI: 10.1002/fld.389)

Fourth-order finite difference simulation of a differentially heated cavity

Hans Johnston*,† and Robert Krasny

Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1109, U.S.A.

SUMMARY

We present benchmark simulations for the 8:1 differentially heated cavity problem, the focus of a special session at the first MIT conference on Computational Fluid and Solid Mechanics in June 2001. The numerical scheme is a fourth-order finite difference method based on the vorticity-stream function formulation of the Boussinesq equations. The momentum equation is discretized by a compact scheme with the no-slip boundary condition enforced using a local vorticity boundary condition. Long-stencil discretizations are used for the temperature transport equation with one-sided extrapolation applied near the boundary. The time stepping scheme for both equations is classical fourth-order Runge–Kutta. The main step is the solution of two discrete Poisson-like equations at each Runge–Kutta time stage, which are solved using FFT-based methods. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: finite difference methods; compact schemes; local vorticity boundary conditions; fast solvers

1. METHODOLOGY

We simulate numerically the 8:1 differentially heated cavity problem, a summary of which can be found in Reference [1]. The flow is governed by the incompressible Boussinesq equations, which when non-dimensionalized and expressed in the vorticity-stream function formulation for a 2D simply connected domain Ω, are given by

$$\theta_t + (\mathbf{u}\cdot\nabla)\theta = \kappa\triangle\theta \tag{1a}$$

$$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \nu\triangle\omega + \theta_x \tag{1b}$$

$$\triangle\psi = -\omega \tag{1c}$$

$$\psi|_\Gamma = 0 \quad \text{and} \quad \partial\psi/\partial n|_\Gamma = 0 \tag{1d}$$

* Correspondence to: H. Johnston, Department of Mathematics, University of Michigan, 4835 E. Hall, Ann Arbor, MI 48109-1109, U.S.A.
† E-mail: hansjohn@umich.edu

Received 31 December 2002
Revised 1 July 2002

where ω, ψ, θ, and $\mathbf{u}$ are the vorticity, stream function, temperature, and velocity field, respectively. The velocity is defined by $\mathbf{u} = (u, v)^{\mathrm{T}} = (\psi_y, -\psi_x)^{\mathrm{T}}$, and Γ denotes the boundary of Ω. Also, $\nu = \sqrt{Pr/Ra}$ and $\kappa = 1/\sqrt{RaPr}$ where Ra is the Rayleigh number and Pr the Prandtl number. For the simulations presented herein, $Ra = 3.4 \times 10^5$ and $Pr = 0.71$. Initial and boundary conditions for the flow have been outlined in an introductory note in this volume and will not be reiterated here.

The numerical method used for our simulations is a fourth-order finite difference scheme, proposed in Reference [2], which we briefly outline. First note that the Boussinesq equations are simply the Navier–Stokes equations supplemented with the temperature equation (1a), which is coupled to the momentum equation (1b) via the gravity term θ_x. The approach taken in Reference [2] is to use an existing Navier–Stokes solver for (1b)–(1d) and then handle the temperature equation separately. This is possible by treating the gravity term explicitly in the time stepping procedure for the system, decoupling the computation of (1a) from that of (1b). In detail, a fourth-order compact discretization, proposed in Reference [3], is used to discretize the momentum equation (1b), including the gravity term. This introduces an auxiliary vorticity variable $\bar{\omega}$ defined in terms of a discrete Poisson-like equation involving ω. Additionally, the kinematic equation (1c) is also discretized using fourth-order compact differencing. The result is again a discrete Poisson-like equation, this time for the stream function with right-hand side $-\bar{\omega}$. This is solved using the Dirichlet boundary condition $\psi|_\Gamma = 0$ by FFT-based methods. Using a discretization of the kinematic relation $\omega = \triangle\psi$ along Γ, the no-slip boundary condition $\partial\psi/\partial n|_\Gamma = 0$ is converted into a local formula for ω along Γ in terms of ψ, i.e., a local vorticity boundary condition; see Reference [4]. The resulting fourth-order approximation is known as Briley's formula. The boundary values of ω provide a Dirichlet boundary condition for the discrete Poisson-like equation for ω, which is solved using FFT-based methods.

The temperature transport equation is treated as a standard convection–diffusion equation. It is discretized using fourth-order long-stencil difference operators, which requires temperature values to be defined at 'ghost' grid points outside of the computational domain. They are prescribed using one-sided extrapolation. To reduce the number of interior points required in the extrapolation, which results in better stability, we apply information obtained from the temperature equation on the boundary.

The classical fourth-order Runge–Kutta method is used for the time discretization of both the momentum equation and the temperature equation. The overall scheme is very efficient. The main step is the solution of two discrete Poisson-like equations at each Runge–Kutta time stage, which are solved using FFT-based methods. Thus, no iterative solvers or stopping criteria are needed. The overall scheme is stable for a given Δt as along as the eigenvalues of system (1), linearized about a fixed velocity, lie within the stability region of the classical fourth-order Runge–Kutta method. This will generally be the case [3] provided that Δt satisfies

$$\frac{\|\mathbf{u}\|_\infty \Delta t}{h} = \mathrm{CFL} \leqslant 1.5 \quad \text{and} \quad \frac{\alpha \Delta t}{h^2} \leqslant \left(\frac{1}{4}\right) \tag{2}$$

where $h = \min\{\Delta x, \Delta y\}$, and $\alpha = \max\{\nu, \kappa\}$. For all of the computations presented here, a fixed Δt was used, dependent on the grid resolution, with $\mathrm{CFL} = 0.75$ and $\|\mathbf{u}\|_\infty = 1.0$.

For metrics involving the pressure we solve the derived pressure Poisson equation

$$\triangle p = \nabla \cdot (-\mathbf{u} \cdot \nabla \mathbf{u} + \hat{\mathrm{j}}\theta) \tag{3}$$

using a second-order finite difference scheme for non-staggered grids proposed in Reference [5]. The Neumann boundary condition

$$\frac{\partial p}{\partial n} = (v \triangle \mathbf{u} + \hat{\mathrm{j}}\theta) \cdot \mathbf{n} \tag{4}$$

is used when solving (3). In order to accurately and consistently approximate this Neumann boundary condition the boundary condition $\nabla \cdot \mathbf{u}|_\Gamma = 0$ from the pressure Poisson equation formulation of the Navier–Stokes equations is used to define the needed 'ghost points' at the discrete level; see Reference [5]. Using this discrete Neumann boundary condition, the pressure Poisson equation (3) is solved using FFT-based methods.

2. RESULTS

Two grid resolutions (N_x, N_y), equi-spaced in each co-ordinate direction, were computed: $(65, 513)$, and $(97, 769)$. Thus, in each case $\Delta x = \Delta y$. The simulations were run until a final non-dimensional time of $T = 1000$. A fixed Δt was used for each simulation determined by (2) with CFL $= 0.75$ and $\|\mathbf{u}\|_\infty = 1.0$.

2.1. Point, wall, and global data

Measurements of the flow variables at 5 data points, defined in Reference [1], were obtained using bicubic interpolation. The derivatives required in the interpolation procedure were approximated to fourth-order accuracy. Amplitudes and periods were computed by finding the zero crossings of mean adjusted measurements using data over 10 periods after $T = 950$. Simpson's rule was used to approximate the integrals in the global metrics.

Tables I and II present results at the two grid resolutions. Point metrics without subscripts are for data point 1, i.e., $(x, y) = (0.181, 7.370)$. Metrics with two subscripts are the time average of the difference between the given variable at the data points indexed by the subscripts; see Reference [1]. Overall the results at the two resolutions are in good agreement. The largest discrepancies are seen in the ΔP pressure measurements. This is most likely attributable to the fact that the pressure is only recovered to second-order accuracy using (3) and (4). However, we emphasize that the pressure computation does not affect the overall accuracy of the numerical scheme since it is performed only to produce the compulsory data.

We point out that in each computation the skewness ε_{12} was identically zero. Thus, the physical skew-symmetry of the non-dimensional temperature θ has been reproduced in our simulation. In Figure 1 is shown a time history of the temperature θ_1 at $(x, y) = (0.181, 7.370)$.

Finally, we note that the data in Tables I and II differs from that presented in Reference [6]. As noted above the metrics in the current work were computed by finding the zero crossings of mean adjusted data. For the results presented in Reference [6] an FFT method was used. Upon examination of the two methods the former was found to be more accurate, which became clear when comparing the metrics under grid refinement. For completeness we include here in Table III the metrics presented in Reference [6] for the finest grid resolution.

Table I. Point and global metrics for 65×513 grid.

	Grid resolution: 65×513 Duration: [950, 984.11], steps per period: 292		
Quantity	Average	Amplitude	Period
Point metrics			
U	5.6551889e − 02	5.6760554e − 02	3.4099372e + 00
V	4.6167337e − 01	7.9432504e − 02	3.4099425e + 00
θ	2.6551460e − 01	4.4197112e − 02	3.4099405e + 00
ε_{12}	0.0000000e + 00	0.0000000e + 00	
ψ	−7.3605769e − 02	7.2565716e − 03	3.4099521e + 00
ω	−2.3583290e + 00	1.1187879e + 00	3.4099067e + 00
ΔP_{14}	−1.1542627e − 03	2.0750962e − 02	3.4099519e + 00
ΔP_{51}	−5.2854594e − 01	2.2864852e − 02	3.4099436e + 00
ΔP_{35}	5.2970020e − 01	1.0362644e − 02	3.4099466e + 00
Global metrics			
$Nu : x = 0$	−4.5801927e + 00	7.2962600e − 03	3.4099466e + 00
$Nu : x = W$	−4.5801927e + 00	7.2962600e − 03	3.4099466e + 00
$\hat{\mathbf{u}}$	2.3953032e − 01	3.4356000e − 05	3.4100230e + 00
$\hat{\omega}$	3.0169977e + 00	3.2773600e − 03	3.4099423e + 00

Table II. Point and global metrics for 97×769 grid.

	Grid resolution: 97×769 Duration: [950, 984.11], steps per period: 438		
Quantity	Average	Amplitude	Period
Point metrics			
U	5.6395438e − 02	5.5246132e − 02	3.4111534e + 00
V	4.6183149e − 01	7.7684964e − 02	3.4111601e + 00
θ	2.6548480e − 01	4.3051242e − 02	3.4111579e + 00
ε_{12}	0.0000000e + 00	0.0000000e + 00	
ψ	−7.3688018e − 02	7.0676312e − 03	3.4111681e + 00
ω	−2.3687941e + 00	1.0875479e + 00	3.4111443e + 00
ΔP_{14}	−1.3921419e − 03	2.0382512e − 02	3.4111650e + 00
ΔP_{51}	−5.3076112e − 01	2.2469632e − 02	3.4111632e + 00
ΔP_{35}	5.3215326e − 01	1.0105539e − 02	3.4111605e + 00
Global metrics			
$Nu : x = 0$	−4.5791427e + 00	7.1351100e − 03	3.4111610e + 00
$Nu : x = W$	−4.5791427e + 00	7.1351100e − 03	3.4111610e + 00
$\hat{\mathbf{u}}$	2.3951401e − 01	3.3738000e − 05	3.4112994e + 00
$\hat{\omega}$	3.0170593e + 00	3.2140900e − 03	3.4111571e + 00

2.2. *Computational resources and timings*

The computations were performed on a single processor DEC Alpha 500au Personal Workstation with 1028 MB of memory and a 2 MB cache. The CPU is an Alpha 21164 500 MHz chip with

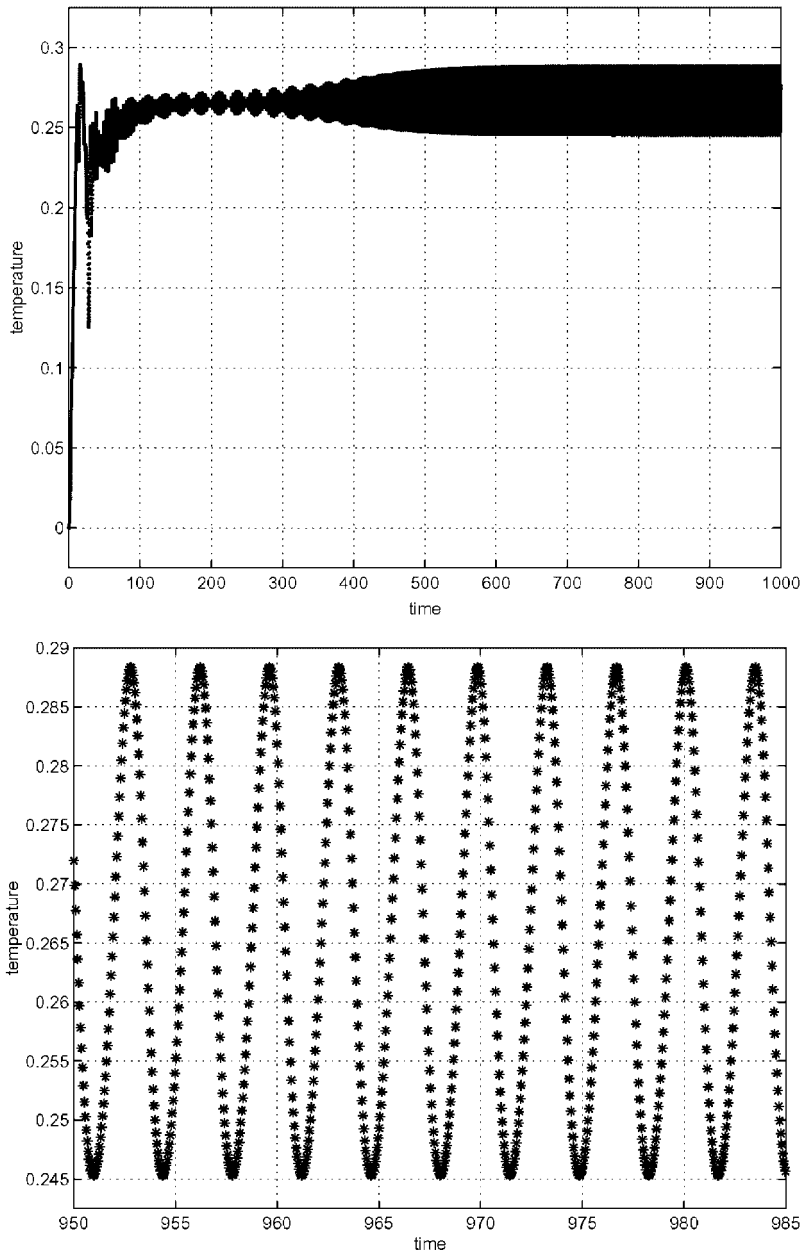

Figure 1. Time history of θ_1, the temperature at $(x, y) = (0.181, 7.370)$ for the $(97, 769)$ grid. (top) full simulation duration; (bottom) eventual oscillatory behaviour of θ_1.

a SPECfp95 rating of 19.5 (18.0 base). All computations were performed in double precision (64 bits), coded in Fortran, and compiled using the *fast* optimization option. Performance data and timings for each grid based on a final simulation time of $T = 1000$ are shown in Table IV.

Table III. Point and global metrics for 97×769 grid (from Reference [6] using FFT method).

	Grid resolution: 97×769 Duration: [950,984.125], steps per period: 438		
Quantity	Average	Amplitude	Period
Point metrics			
U	5.616e − 02	5.452e − 02	3.422e + 00
V	4.605e − 01	7.718e − 02	3.422e + 00
θ	2.647e − 01	4.268e − 02	3.422e + 00
ε_{12}	0.000e + 00	0.000e + 00	
ψ	−7.348e − 02	6.856e − 03	3.422e + 00
ω	−2.362e + 00	9.940e − 01	3.422e + 00
ΔP_{14}	−1.375e − 03	2.004e − 02	3.422e + 00
ΔP_{51}	−5.293e − 01	2.236e − 02	3.422e + 00
ΔP_{35}	5.307e − 01	1.013e − 02	3.422e + 00
Global metrics			
$Nu : x = 0$	−4.567e + 00	7.130e − 03	3.422e + 00
$Nu : x = W$	−4.567e + 00	7.130e − 03	3.422e + 00
$\hat{\mathbf{u}}$	2.389e − 01	3.366e − 05	3.422e + 00
$\hat{\omega}$	3.001e + 00	3.216e − 03	3.422e + 00

Table IV. Timings and performance data.

Grid size	Δt	Steps	CPU (sec)	Memory (MB)	CPU (μs)/point/step
(65, 513)	1.171875e − 02	85 333	28 691	7.02	10.08
(97, 769)	7.812500e − 03	128 000	121 161	11.41	12.69

As noted above the simulations were performed using a fixed Δt. This was done in order to collect flow data at a precise fixed time interval, and allowed FFT techniques to be used to compute the metrics presented in the conference proceedings [6]. As noted above the results presented in this work compute the metrics by determining the zero crossings of mean adjusted data, which we found to be more accurate. Thus, a fixed Δt is not necessary and in fact adds significantly to the total computational time. This was determined by performing a simulation and allowing for a variable time step. We took $(N_x, N_y) = (97\,769)$ and CFL = 1.0. Allowing for a variable time step by monitoring $\|\mathbf{u}\|_\infty$ during the run to determine Δt resulted in a one-third reduction in the total run time; see Reference [2].

3. CONCLUSION

Benchmark quality results of the simulation of a 8:1 differentially heated cavity were presented. The numerical method used is fully explicit in time, simple to implement, and highly efficient. The implementation of the spatial discretization and time stepping make possible the solution of the linear systems by direct FFT-based solvers. The resulting efficiency makes the method well suited for numerical studies at moderate to large Rayleigh numbers, orders of magnitude

above that studied in this work. In this regime the convective time constraint would be more restrictive than the diffusive time constraint. This is the best that can be hoped for in the sense that in practice one must always honour the convective time step constraint in order to faithfully follow the dynamics of the flow [2].

REFERENCES

1. Christon MA, Gresho PM, Sutton SB. Computational predictability of natural convection flows in enclosures. *International Journal for Numerical Methods in Fluids* 2002; **40**:953–980.
2. Liu J-G, Wang C, Johnston H. A fourth-order compact scheme for incompressible Boussinesq equations. *Journal of Scientific Computing* 2003; at press.
3. WE, Liu J-G. Essentially compact schemes for unsteady viscous incompressible flows. *Journal of Computational Physics* 1996; **126**:122–138.
4. WE, Liu J-G. Vorticity boundary condition and related issues for finite difference schemes. *Journal of Computational Physics* 1996; **124**:368–382.
5. Johnston H, Liu J-G. Finite difference schemes for incompressible flow based on local pressure boundary conditions. *Journal of Computational Physics* 2002; **180**:120–154.
6. Johnston H, Krasny R. Computational predictability of natural convection flows in enclosures: A benchmark problem. In *Computational Fluids and Solid Mechanics*, Bathe KJ (ed.). Elsevier Science: Amsterdam, 2001.

