Pergamon

Int. J. Heat Mass Transfer. Vol. 39, No. 12, pp. 2401–2407, 1996

0017-9310/96 $15.00 + 0.00

0017-9310(95)00346-0

A comparison of the reduced Galerkin and pseudo-spectral methods for simulation of steady Rayleigh–Bénard convection

L. E. HOWLE

Department of Mechanical Engineering and Materials Science, Box 90300 Hudson Hall, Duke University, Durham, NC 27708-0300, U.S.A.

(*Received* 27 *March* 1995 *and in final form* 13 *September* 1995)

Abstract—A study of the computational efficiency of two numerical methods based on a mixed finite difference–Galerkin technique is undertaken. This study uses steady Rayleigh–Bénard convection in a periodic container as a model problem. The formulation and linearization of the reduced Galerkin and pseudo-spectral methods is discussed. A new technique for reducing the computational effort of evaluating the convolution sums is used. It is found that the reduced Galerkin method allows greater linearization of the equations of fluid motion. Additionally, the reduced Galerkin method is approximately three times faster than the pseudo-spectral method for the problem studied. Copyright © 1996 Elsevier Science Ltd.

1. INTRODUCTION

The study of buoyancy-driven fluid motion is important to many areas of the scientific community from geology and meteorology to heat transfer and nonlinear dynamics. For example, motion inside the Earth's mantle is caused by buoyancy effects. It has recently been demonstrated that internal heating due to viscous dissipation can significantly alter the convection patterns [1]. In the heat transfer community, natural convection is an extensively studied topic [2]. In the nonlinear dynamics field, certain natural convection problems represent important pattern forming systems [3] and containing rich bifurcation structure [4]. Rayleigh–Bénard convection is a paradigm of this class of problems and has been extensively studied [5]. For this reason the numerical methods explored in this paper will use two-dimensional (2D) steady Rayleigh–Bénard convection as a model problem.

In the study of fluid dynamics, numerical computation is a widely used and important tool. The numerical solution provides local information which can be very expensive or impossible to obtain using experimental methods [1, 6]. Several numerical methods are commonly in use. These include the control volume [7], finite difference [8], finite element [9] and spectral methods [10], among others. For some problems, spectral methods offer computational advantages over other methods [11].

When the equations which describe the physical system are nonlinear or contain non-constant coefficients spectral methods (for example Galerkin methods) are impractical, except for cases where very low spatial resolution is required [12]. This is because the majority of the computational effort is required to evaluate the forcing vector, **b**, of the linearized system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{1}$$

For the Galerkin formulation of nonlinear flow problems, **b** contains the convolution sums which result from using global approximations (Fourier series, Chebyshev polynomials, surface harmonics, etc.) with quadratic non-linearities [13]. When pseudo-spectral methods were devised, spectral methods become competitive with other computational methods [14, 15].

In this work, I present an acceleration method which is performed on the Galerkin equations. This reduced method, which was first reported by Howle [16], is executed entirely in the spectral space. The reduced Galerkin method (RGM) exploits the sparse nature of the convolution products by expanding the convolution sums and eliminating terms which are multiplied by zero elements of the convolution products. This rather simple idea results in tremendous computational savings over traditional Galerkin methods. Additionally, the RMG is approximately three times faster than the pseudo-spectral method (PSM) for the problem used in this paper.

Expansion of the convolution sums is accomplished by a source-code-writing utility. This is a computer program which writes the expanded and reduced convolution sums to source code files. The coding utility also writes the necessary header files and the Makefile. This makes the convolution sum expansion transparent to the user. The program needs to be re-compiled only when the spectral truncation level is changed.

The following sections include the formulation of

NOMENCLATURE

A	coefficient matrix, equation (1)
b	forcing vector, equation (1)
D	partial derivative w.r.t. z
$\mathbf{e}_z$	vertical basis vector
f_1, f_2	convolution summations, equations (17) and (19)
g	acceleration of gravity
K, L	Fourier truncation level
p	pressure
Ra	Rayleigh number, equation (5)
Pr	Prandtl number, equation (6)
t	time
T	temperature
v, w	horizontal, vertical velocity components
V	velocity vector
x	solution vector, equation (1).

Greek symbols

β	coefficient of thermal expansion
δ_T	relaxation parameter
δ_{mn}	Kronecker delta
κ	thermal diffusivity
λ	aspect viscosity
ν	kinematic viscosity
ζ	vorticity, equation (8).

Subscripts

k, l, m kth, lth, mth mode.

the weak form of the Galerkin ordinary differential equations, discussion of the linearization and reduction, derivation of the PSM and a comparison of the convergence behavior and computation time of the two methods.

2. GALERKIN FORMULATION

For the horizontal fluid layer shown in Fig. 1, the equations describing the conservation of mass, momentum, and energy are

$$\nabla \cdot \mathbf{V} = 0 \tag{2}$$

$$\frac{1}{Pr}\left(\frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V}\right) = -\nabla p^* + Ra T \mathbf{e}_z + \nabla^2 \mathbf{V} \tag{3}$$

and

$$\frac{\partial T}{\partial t} + \mathbf{V} \cdot \nabla T = \nabla^2 T \tag{4}$$

where p^* is the pressure perturbation, $p^* = p - p_{\text{hydrostatic}}$. The dimensionless equations (2)–(4) are subject to no-slip isothermal boundary conditions and the Boussinesq approximation is assumed to hold. The Rayleigh and Prandt numbers in equation (3) are defined by

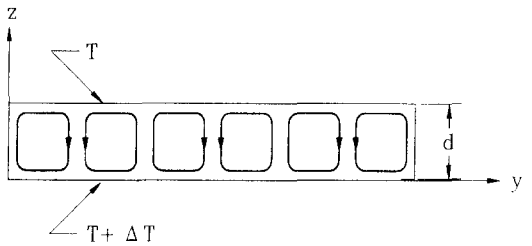

Fig. 1. An unstably heated horizontal fluid layer.

$$Ra = \frac{\beta g \Delta T d^3}{\kappa \nu} \tag{5}$$

and

$$Pr = \frac{\nu}{\kappa}. \tag{6}$$

Here, β is the isobaric thermal expansion coefficient, ΔT is the temperature difference across the fluid layer, d is the layer height, κ is the thermal diffusivity and ν is the kinematic viscosity.

Pressure is eliminated by taking the curl of the momentum equation (3). The resulting conservation form of the vorticity transport equation is

$$\frac{1}{Pr}\left(\frac{\partial \zeta}{\partial t} + \frac{\partial (v\zeta)}{\partial y} + \frac{\partial (w\zeta)}{\partial z}\right) = Ra \frac{\partial T}{\partial y} + \nabla^2 \zeta \tag{7}$$

where the vorticity

$$\zeta = \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \tag{8}$$

with v and w denoting the horizontal and vertical velocity components, respectively. We will retain v and w in equation (7) rather than use a stream function. This requires the use of the weak form of the divergence-free condition to enforce mass conservation. Equation (8) is substituted into equation (7) and time dependent terms are dropped for the derivations which follow.

Because of the problem's symmetry and expected steady roll patterns [17] at the moderate Rayleigh numbers used in these computations, the T and w fields are represented [18] by even Fourier series

$$T(y, z) \approx T_0(z) + \sqrt{2} \sum_{k=1}^{K} T_k(z) \cos(\alpha_k y) \tag{9}$$

and

$$w(y,z) \approx \sqrt{2}\sum_{k=1}^{K} w_k(z)\cos(\alpha_k y) \tag{10}$$

while the horizontal velocity, v, is given by an odd Fourier series

$$v(y,z) \approx \sqrt{2}\sum_{k=1}^{K} v_k(z)\sin(\alpha_k y) \tag{11}$$

where

$$\alpha_k = \frac{2\pi k}{\lambda} \tag{12}$$

and λ is the fluid layer aspect ratio (width/height). The modal amplitudes in equations (9)–(11) are functions of z only and derivatives with respect to z are approximated by centered, second-order finite differences. Other trial functions are commonly used. For example, Thess and Orszag [6] use a Fourier expansion in the horizontal directions and Chebyshev polynomials in the vertical direction for simulation of 3D Bénard–Marangoni convection.

In the spectral space, the orthogonal basis functions

$$1, \sqrt{2}\sin(\alpha_m y), \sqrt{2}\cos(\alpha_m y) \tag{13}$$

are defined. The inner product

$$\langle f, b\rangle = \frac{1}{\lambda}\int_0^{\lambda} fb\,\mathrm{d}y \tag{14}$$

projects the trial functions onto the basis.

A detailed derivation of the weak form of the Galerkin ordinary differential equations for this problem is given in Howle [16] and will not be repeated here. The zeroth temperature modal amplitude is given by

$$D^2 T_0 = \sum_{m=1}^{K} (Dw_m T_m + w_m DT_m) \tag{15}$$

where D denotes the derivative with respect to the vertical direction, z. The higher temperature modes are given by the K equations

$$D^2 T_m - (\zeta w_{2m})DT_m - \left(\alpha_m^2 + \frac{\zeta}{2}Dw_{2m}\right)T_m = DT_0 + \zeta(2T_{2m}Dw_m + DT_{2m}w_m) + \zeta f_1 \tag{16}$$

with the convolution summation defined as

$$f_1 = \sum_{k=1\neq m}^{K}\sum_{l=1\neq m}^{K}\left(\frac{\alpha_k T_k Dw_l}{\alpha_l} I_4 + DT_k w_l I_1\right) \tag{17}$$

where $\zeta = 1/\sqrt{2}$.

The weak Galerkin ordinary differential equations for the vertical velocity modal amplitudes are

$$D^4 w_m + (\phi w_{2m})D^3 w_m + \left(\frac{\phi}{2}Dw_{2m} - 2\alpha_m^2\right)D^2 w_m + (\alpha_{2m}^2\phi w_{2m} - \phi D^2 w_{2m} - \alpha_m^2\phi w_{2m})Dw_m + \left(\alpha_m^4 + \alpha_m\alpha_{2m} - \frac{\phi}{2}D^3 w_{2m} - \frac{\alpha_m^2}{2}\phi Dw_{2m}\right)w_m = \alpha_m^2 Ra T_m + \phi f_2 \tag{18}$$

with the convolution sums

$$f_2 = \sum_{k=1\neq m}^{K}\sum_{l=1\neq m}^{K}\left(\left(\frac{\alpha_k^2 w_k Dw_l}{\alpha_l} - \frac{D^2 w_k w_l}{\alpha_l}\right)I_2 + \left(\frac{D^3 w_k w_l}{\alpha_l} - \alpha_k Dw_k w_l\right)I_3\right) \tag{19}$$

and

$$\phi = \frac{1}{\sqrt{2\,Pr}}. \tag{20}$$

The convolution products in equations (17) and (19) are

$$I_1 = \delta_{mn} + \delta_{mo} + \delta_{mp} \tag{21}$$

$$I_2 = \delta_{mn} - \delta_{mo} + \delta_{mp} \tag{22}$$

$$I_3 = \delta_{mn} + \delta_{mo} - \delta_{mp} \tag{23}$$

$$I_4 = \delta_{mn} + \delta_{mo} + \delta_{mp} \tag{24}$$

where $n = k+l$, $o = k-l$ and $p = l-k$.

Equations (16) and (18) are linearized by extracting the $l = m$ and $k = m$ terms from the convolution sums, equations (17) and (19). Terms containing T_m and its derivatives are moved from (17) to the left-hand side of (16). Likewise, terms containing w_m and its derivatives are moved from (19) to the left-hand side of (18). This accounts for the indexing $k = 1 \neq m$ and $l = 1 \neq m$ in equations (17) and (19). This quasi-linearization [19] is expected to increase the solution convergence rate.

The improvements in computational efficiency of the reduced Galerkin method result from reducing the effort in evaluating the convolution sums in (17) and (19) as discussed in Howle [16]. The orthogonality of the trial functions [equations (9), (10) and (11)] with the basis [equation (13)] causes the convolution products [equations (21)–(24)] to be sparse. If the convolution sums are evaluated directly, this sparseness can not be exploited. An alternative to direct evaluation of the convolution sums is to expand the sums for a given truncation level, K. For example, suppose the desired truncation level is $K = 4$ and we are solving for the $m = 3$ temperature modal amplitude, T_3. Expanding equation (17) gives

$$\begin{aligned} f_1 = {} & T_1 Dw_1 I_4 + DT_1 w_1 I_1 + \tfrac{1}{2}T_1 Dw_2 I_4 + DT_1 w_2 I_1 \\ & + \tfrac{1}{4}T_1 Dw_4 I_4 + DT_1 w_4 I_1 + 2T_2 Dw_1 I_4 + DT_2 w_1 I_1 \\ & + T_2 Dw_2 I_4 + DT_2 w_2 I_1 + \tfrac{1}{2}T_2 Dw_4 I_4 + DT_2 w_4 I_1 \\ & + 4T_4 Dw_1 I_4 + DT_4 w_1 I_1 + 2T_4 Dw_2 I_4 + DT_4 w_2 I_1. \end{aligned} \tag{25}$$

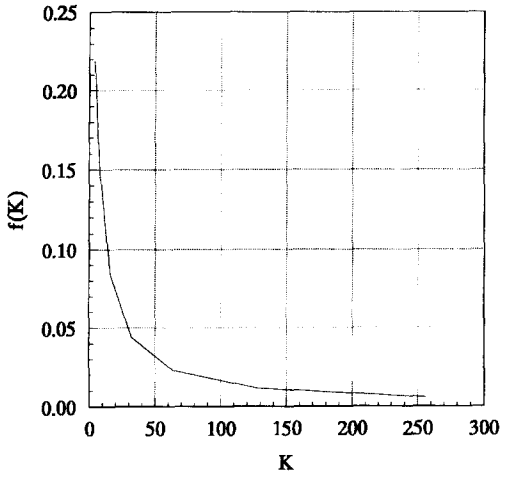

Fig. 2. Fraction of non-zero components of the convolution product, I_1, vs truncation level, K.

Using the convolution products, equations (21)–(24), this is reduced to

$$f_1 = -\tfrac{1}{2}T_1 Dw_2 + DT_1 w_2 + \tfrac{1}{4}T_1 Dw_4 + DT_1 w_4 - 2T_2 Dw_1 + DT_2 w_1 + 4T_4 Dw_1 + DT_4 w_1. \quad (26)$$

As the truncation level, K, is increased, a greater percentage of the terms in the convolution sums are eliminated. This point is made graphically by Fig. 2 which shows the fraction of non-zero elements of I_1 vs K. The remaining convolution products are equally sparse. Exploiting this sparsity by expanding and reducing the convolution sums as just outlined can result in significant computational savings. In [16], convolution sum reduction was shown to reduce both computation time and memory usage by orders of magnitude when compared to the Galerkin–finite difference method.

To make the reduction practical, the convolution sums must be evaluated and the source code written by computer. To do this, I wrote a short program (three pages of C code) which writes a function source code file for each of the K equations (17) and (19). Alternatively, the source code can be included in the calling routine by using the proper preprocessor directives. To make compiling easier, the code generation program also writes the necessary header files and the Makefile. The entire convolution sum evaluation and source code generation process is transparent to the user.

3. PSEUDO-SPECTRAL FORMULATION

For the pseudo-spectral formulation, the nonlinear terms in equations (7) and (4) are themselves represented by series expansions. Equation (8) is substituted into the second and third terms on the left-hand side of equation (7) and approximated by the series

$$v\frac{\partial}{\partial y}\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) + w\frac{\partial}{\partial z}\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right) \approx \sqrt{2}\sum_{k=1}^{L} A_k(z)\sin(\alpha_k y). \quad (27)$$

Likewise, the second and third terms in equation (4) are written as

$$v\frac{\delta T}{\delta y} + w\frac{\delta T}{\delta z} \approx B_0(z) + \sqrt{2}\sum_{k=1}^{L} B_k(z)\cos(\alpha_k y). \quad (28)$$

The series in equations (27) and (28) preserve the spatial symmetries [which are imposed by the trial functions, equations (9)–(11)] of the terms they represent.

The rest of the formulation of the weak form of the pseudo-spectral case proceeds as with the Galerkin case. The zeroth mode temperature equation is

$$D^2 T_0 = B_0 \quad (29)$$

while the remaining temperature modal amplitudes are found by solving the equations

$$D^2 T_m - \alpha_m^2 T_m = B_m. \quad (30)$$

The vertical velocity modal amplitudes are given by the equations

$$(D^2 - \alpha_m^2)^2 W_m = \alpha_m Pr^{-1} A_m + \alpha_m^2 Ra T_m. \quad (31)$$

The boundary conditions for both the Galerkin and pseudo-spectral formulations are

$$T_0^{z=0} = 1$$

$$T_0^{z=1} = T_m^{z=0} = T_m^{z=1} = w_m^{z=0} = w_m^{z=1} = 0. \quad (32)$$

From the no-slip condition and the weak form of the continuity equation

$$v_m = -\frac{Dw_m}{\alpha_m} \quad (33)$$

the two additional boundary conditions required for equations (18) and (31) are

$$Dw_m^{z=0} = Dw_m^{z=1} = 0. \quad (34)$$

Additionally, this weak form of the continuity equation (33) is used to eliminate v_m from all of the Galerkin and pseudo-spectral ordinary differential equations and thus enforces the conservation of mass.

As discussed in [14, 15], the pseudo-spectral computations are performed partially in spectral space and partially in physical space. As equation (29), the K equations (30), and the K equations (31) are solved, the new modal amplitudes T_0, T_m and w_m are inverse transformed so that the temperature and vertical velocity fields are known in the physical space. The z derivatives of the vertical velocity amplitudes are used with equation (33) to compute the horizontal velocity amplitudes. These are inverted so the horizontal vel-

ocity is also known in the physical space. The nonlinear terms [left-hand sides of equations (28) and (27)] are calculated in physical space where the information is local rather than global. They are then transformed to spectral space to find the new estimates for the modal amplitudes A_m, B_0 and B_m. A fast transform process [20] is used which requires only $O(N\log_2 N)$ operations. The new model amplitudes are used for the next iteration and the process is repeated until the solution converges.

By examining the left-hand sides of equations (27) and (28), it is evident that the number of modes, L, used to represent the nonlinear terms must be $2K$ or greater in order to prevent aliasing. Further, by the sampling theorem, a minimum of $2L$ (or $4K$) collocation points must be used for physical space representation of these terms. For the results given in the next section, $2K$ modes are used to represent the nonlinear terms in the spectral space and $4K$ collocation points are used in the physical space. It should be noted that phase shifting is an alternative method for preventing aliasing [21, 22].

4. RESULTS AND DISCUSSION

Both the reduced Galerkin and the pseudo-spectral methods must be under-relaxed in order to prevent divergence. The under relaxation is dependent upon a number of factors, two of which are the Rayleigh number and the quality of the initial guess. For both methods, relaxation of the vertical velocity amplitudes is unnecessary. The temperature modal amplitudes, on the other hand, are under relaxed as

$$T_m^{\text{new}} = \delta_T T_m^{\text{new}} + (1-\delta_T) T_m^{\text{old}} \tag{35}$$

where $0 \leqslant \delta_T \leqslant 1$. Because the two methods are linearized to a different degree, it is reasonable to expect the convergence rate to differ. As shown in Fig. 3, the optimal value of the relaxation parameter for the two schemes is different. The figure shows the number of iterations required for solution convergence vs the relaxation parameter, δ_T. In both cases, an identical initial guess is used and a solution is found for $Ra = 5000$, $Pr = 6$ and an aspect ratio $\lambda = 4.0$. These parameters give typical results. The *relative* performance of the two methods changes little with other parameter values. For the results which follow, each of the computational methods uses its optimal δ_T which at this Ra and Pr are 0.69 for the RGM and 0.15 for the PSM. The figure shows that the RGM converges in 35 iterations while the PSM takes about 190 iterations at optimal damping.

Another desirable feature of the RGM convergence behavior is the curve shape. Since the optimal damping parameter value must be found by trial and error, the shallow slope of the iteration vs δ_T curve over a large δ_T range makes the choice of the damping parameter less critical for the RGM. Figure 3 shows that the PSM has a fairly narrow range of 'acceptable' δ_T. The RGM, on the other hand, is less sensitive to the value of δ_T so long as the value is less than the optimal.

Convergence is achieved when a suitable norm stops changing. For both of the present methods the max norm of the amplitudes is used. For example,

$$\max_{0\leqslant k\leqslant K} \max_{1\leqslant n\leqslant NZ} (|w_{k,n}^i - w_{k,n}^{i-1}|, |T_{k,n}^i - T_{k,n}^{i-1}|) \leqslant \varepsilon \tag{36}$$

where NZ is the number of finite-difference points in the vertical direction and i is the iteration. The values $\varepsilon = 10^{-4}$ and $NZ = 51$ are used for this work.

Figure 4 shows the execution time vs truncation level for the two methods. While the execution time per iteration for the PSM is less than the execution time per iteration for the RGM, the number of iterations required for convergence is significantly less for the RGM. This is because the RGM is linearized to a

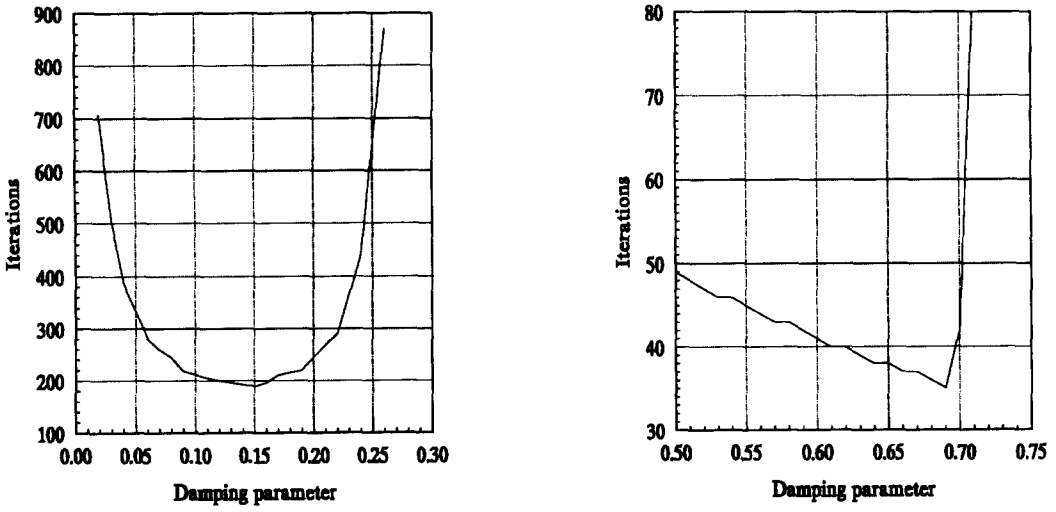

Fig. 3. Number of iterations required for convergence for the pseudo-spectral (left) and reduced Galerkin (right) methods.

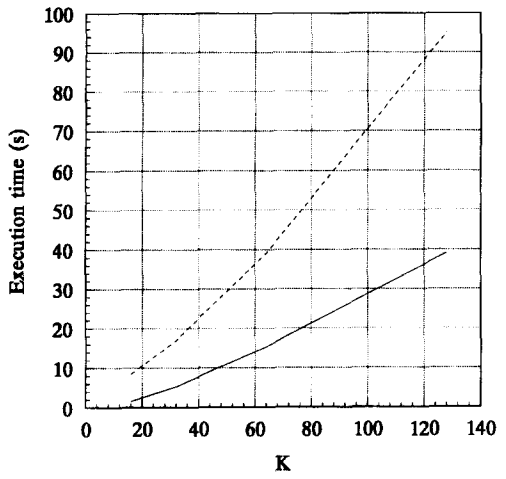

Fig. 4. Execution time of the pseudo-spectral (upper curve) and reduced Galerkin (lower curve) methods.

greater degree than the PSM. As a result, the RGM requires less execution time to converge to a solution.

The time required to compile the RGM program can be considerable when the Fourier truncation level is large. The 128 mode case requires several hours of compile time on an IBM RS/6000 model 370. This is because the number of lines of source code for the reduced convolution sums is $6K^3f(K)$ where $f(K)$, the fraction of non-zero elements of the convolution products, is shown in Fig. 2. Here, each term is placed on a single line so that equation (26) would contain eight lines. The total number of lines of code for both convolution sums is plotted in Fig. 5. For a specific example, the RGM $K = 32$ program has 8736 lines of code whereas the $K = 128$ case requires 148 536 lines of code for the expanded and reduced convolution sums. The total memory required for the RGM is greater than with the PSM but is much less than with the Galerkin method [16].

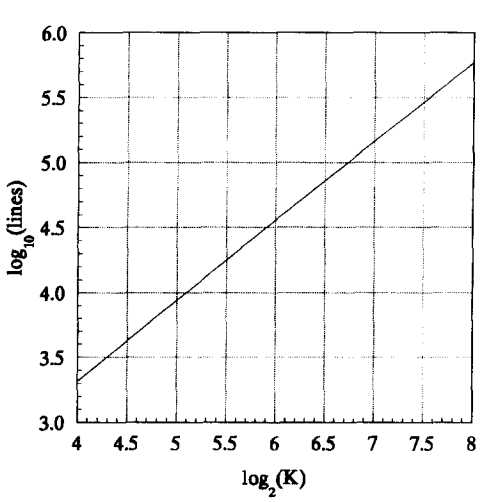

Fig. 5. Number of source code lines for the expanded and reduced convolution sums.

5. CONCLUSION

For the case of steady, 2D Rayleigh–Bénard convection, I have demonstrated a new method, the reduced Galerkin method (RGM), for accelerating convolution sum evaluation. Unlike the pseudo-spectral method (PSM), the RGM is computed entirely in the spectral space. The RGM is linearizable to a greater degree than the PSM. As a result, the RGM requires fewer iterations for convergence than the PSM.

The RGM has the disadvantage of more involved mathematical formulation. For problems which do not need high spatial resolution, the increase in execution speed of the RGM is probably not worth the extra formulation and coding effort. Conversely, for high resolution problems, the RGM offers significant computational savings over PSM.

The essential tool for the successful use of the RGM is a coding program. This coding program writes the expanded and simplified convolutions sum source code files. Without the use of a code writing program, the RGM formulation is too labor intensive to be of any practical use.

Acknowledgements—I would like to gratefully acknowledge the support of the Mechanical Engineering and Materials Science Department and the School of Engineering at Duke University.

REFERENCES

1. S. Balachandar, D. A. Yuen, D. M. Reuteler and G. S. Lauer, Viscous dissipation in three-dimensional convection with temperature-dependent viscosity, *Science* **267**, 1150–1153 (1995).
2. A. Bejan, *Convection Heat Transfer*. John Wiley, New York (1984).
3. M. Seul and D. Andelman, Domain shapes and patterns: the phenomenology of modulated phases, *Science* **267**, 476–483 (1995).
4. J. Gollub and S. V. Benson, Many routes to turbulent convection, *J. Fluid Mech.* **100**, 449–470 (1980).
5. M. C. Cross, Ingredients of a theory of convective textures close to onset, *Phys. Rev. A* **25**, 1065–1076 (1982).
6. A. Thess and S. A. Orszag, Surface-tension-driven Bénard convection at infinite Prandtl number, *J. Fluid Mech.* **283**, 201–230 (1995).
7. D. Mukutmoni and K. T. Yang, Thermal convection in small enclosures: an atypical bifurcation sequence, *Int. J. Heat Mass Transfer* **38**, 113–126 (1995).
8. J. O. Wilkes and S. W. Churchill, The finite difference computation of natural convection in a rectangular enclosure, *A.I.Ch.E. Jl* **12**, 161–166 (1966).
9. V. D. Murty, A numerical investigation of Bénard convection using finite elements, *Comput. Fluids* **14**, 379–391 (1986).
10. L. E. Howle and J. G. Georgiadis, Natural convection in porous media with anisotropic dispersive thermal conductivity, *Int. J. Heat Mass Transfer* **37**, 1081–1094 (1994).
11. S. A. Orszag, Numerical simulation of incompressible flows within simple boundaries—I. Galerkin (spectral) representations, *Studies Appl. Maths* **L**, 293–327 (1971).
12. C. A. J. Fletcher, *Computational Galerkin Methods* (Edited by H. Cabannes, M. Holt, H. B. Keller, J. Killeen and S. A. Orszag). Springer, New York (1984).
13. C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A.

Zang, *Spectral Methods in Fluid Flow* (Edited by R. Glowinski, M. Holt, P. Hut, H. B. Keller, J. Killeen, S. A. Orszag and V. V. Rusanov). Springer, New York (1988).
14. S. A. Orszag, Numerical methods for the simulation of turbulence, *Phys. Fluids* **12** (Suppl. 2), 250–257 (1969).
15. S. A. Orszag, Transform method for calculation of vector coupled sums: Application to the spectral form of the vorticity equation, *J. Atmos. Sci.* **27,** 890–895 (1970).
16. L. E. Howle, Efficient implementation of a finite difference Galerkin method for simulation of large aspect ratio convection, *Numer. Heat Transfer B* **26,** 105–114 (1994).
17. S. Chandrasekher, *Hydrodynamic and Hydromagnetic Stability*. Oxford University Press, Oxford (1967).
18. J. M. McDonough and I. Catton, Mean field equations of Bénard convection, *Phys. Fluids* **23,** 1064–1065 (1980).
19. J. M. McDonough and I. Catton, A mixed finite difference–Galerkin procedure for 2D convection in a square box, *Int. J. Heat Mass Transfer* **25,** 1137–1146 (1982).
20. J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, *Math. Comput.* **19,** 297–301 (1965).
21. R. S. Rogallo, An ILLIAC program for the numerical simulation of homogeneous incompressible turbulence, NASA TM-73203 (1977).
22. G. S. Patterson and S. A. Orszag, Spectral calculation of isotropic turbulence: efficient removal of aliasing interaction, *Phys. Fluids* **14,** 2538–2541 (1971).