Natural convection in a sidewall heated cube using an immersed boundary method

K. A. Dittko[1] M. P. Kirkpatrick[2] S. W. Armfield[3]

(Received 31 January 2011; revised 4 July 2011)

Abstract

Numerical simulation of a square cavity was conducted to validate an implementation of an Immersed Boundary Method (IBM). The cavity consists of two differentially heated side walls and adiabatic top and bottom walls. A Cartesian grid is used with a finite volume, fractional step pressure correction method. Simulations use Dirichlet boundaries for vertical walls and Neumann boundaries for horizontal walls. The Immersed Boundary Method involves modifying the Navier–Stokes equations to include a forcing function in the momentum and energy equations that creates a virtual boundary. This method is useful because the boundary does not necessarily have to coincide with grid points; however, it is much less computationally expensive than other similar methods such as the cut cell method. The IBM is commonly used in simulations involving complex objects and can also accommodate moving boundaries. A standard numerical simulation with grid aligned with the boundary is first compared with previous results. The same geometry is then simulated by tilting the grid at various angles and

http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/3967 gives this article, © Austral. Mathematical Soc. 2011. Published August 1, 2011. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

using the IBM for each of the walls, and comparing these results with those initially obtained. We detail the implementation method and common problems associated with this. Velocity and temperature profiles are presented and the IBM is shown to maintain second order spatial accuracy.

Contents

1 Introduction

The Immersed Boundary Method (IBM) has been used successfully for a number of applications. The IBM allows simulation of complex geometry, including moving boundaries, on a Cartesian grid. An example of an early implementation is by Peskin [7] who simulated blood flow through a human heart. This is particularly interesting as the work involved complex geometry as well as moving boundaries.

Simulation of flow past a cylinder is commonly used as a way of testing a particular IBM implementation [1, 2, 9]. These examples all looked at the momentum equation aspect of the boundary; however, treatment of heat transfer at the boundary is much less common. One example that does examine heat transfer is the work of Zhang and Zheng [11]. Both Dirichlet and Neumann type boundaries were simulated, but again concentrating on flow past a cylinder.

This article concentrates on the heat transfer aspect of the IBM in natural convection flow rather than the traditional forced convection cases typically shown. This is done by simulating a cube with differentially heated side-walls. This problem has been studied extensively both experimentally and numerically [8, 6]; however, it has not been studied in conjunction with the IBM.

2 Method

Figure 1 illustrates the configuration of the computational grid and the flow domain. The hat operator (for example $\hat{X}$) represents values associated with the computational grid coordinates, while other values are associated with the flow domain coordinates. The angle between the two sets of coordinate axes is γ. All work presented is 2D, therefore the z axes of each coordinate system are always aligned. Extension to 3D is straightforward.

The area of interest is the shaded area in Figure 1, with the surrounding area required to enforce the boundary. This surrounding area is simulated but has no effect on the shaded area we are interested in so is simply ignored. The result is a square domain with the grid sloped relative to the walls and gravity (the gravity vector $\mathfrak{g}$ is always aligned with the $\mathfrak{y}$ direction).

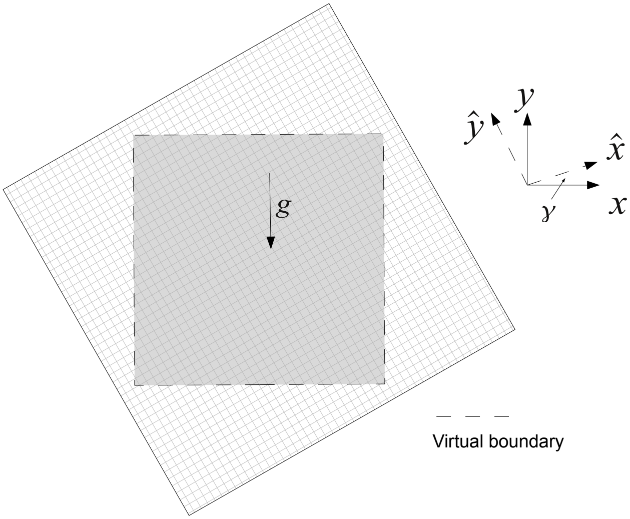

Figure 1: This is the setup of the grid. Dashed lines represent virtual boundaries, with the shaded area being the part of the simulation that is of interest. The hat operator represents values in the coordinate system aligned with the grid, as opposed to the coordinate system of the tilted domain.

2.1 Immersed boundary method

This section consists of a brief overview of the IBM used. The method used for the Immersed Boundary is based on work by Zhang et al. [10, 11], and these articles contain a more comprehensive description of the procedure.

The original solver used was originally developed and validated by Kirkpatrick et al. [3, 4]. It uses a finite volume, fractional step, pressure correction method. The code solves the Navier–Stokes equations using a fourth order Central Difference scheme for momentum, a fourth order Central Difference scheme with ULTRA flux limiter [5] for scalars and second order accurate time stepping.

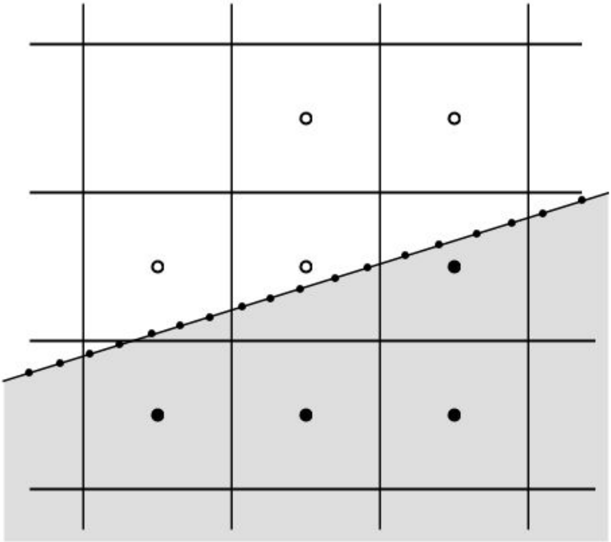

Figure 2: Diagram showing the grid in the vicinity of the Immersed Boundary. The small black circles are the virtual boundary points and have coordinates $\mathbf{x}_s$. The large filled circles below the virtual boundary represent points on the computational grid outside the flow where the forcing functions are applied. The large rings on the other side of the boundary are in the flow, meaning the forcing function is not applied, but they are used to calculate the properties at the virtual boundary points.

The IBM was implemented within this code through the addition of an extra term $\mathbf{f_m}$ to the momentum equations, and an extra term f_e to the energy equation, so the equations solved become

$$\nabla \cdot \mathbf{u} = 0\,, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho_{\text{ref}}} \nabla P + \frac{\rho - \rho_{\text{ref}}}{\rho_{\text{ref}}} \mathbf{g} + \nu \nabla^2 \mathbf{u} + \mathbf{f_m}\,, \tag{2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \frac{\nu}{\Pr} \nabla^2 T + f_e\,, \tag{3}$$

where $\mathbf{u}$ is the velocity vector, ρ is density, ρ_{ref} is density at the reference conditions, P is pressure, $\mathbf{g}$ is the gravity vector, ν is kinematic viscosity, T is temperature, and Pr is the Prandtl number.

The forcing terms are calculated by comparing actual velocity and temperature values on the boundary to desired values. There are nodes situated on the Immersed Boundary as shown in Figure 2. For any given Immersed Boundary node the actual value at the node is calculated using a weighted average of the four surrounding grid nodes. This procedure uses the equations

$$\mathbf{U}(\mathbf{x}_s) = \sum_{i_s,j_s}^{i_s+1,j_s+1} D_{i,j}(\mathbf{x}_s)\mathbf{u}_{i,j} \quad \text{and} \quad T(\mathbf{x}_s) = \sum_{i_s,j_s}^{i_s+1,j_s+1} D_{i,j}(\mathbf{x}_s)T_{i,j}\,, \tag{4}$$

where $\mathbf{x}_s$ is the position of the nodes, (i_s, j_s) is the grid position immediately below and left of the Immersed Boundary node and the weighting function

$$D_{i,j}(\mathbf{x}_s) = d(x_s - x_i)d(y_s - y_i). \tag{5}$$

Here (x_s, y_s) is the position of the node on the Immersed Boundary whereas (x_i, y_i) is the position of a node on the computational grid. This weighting function is non-zero only when the boundary point is within one grid space of the grid node of interest. The terms on the right hand side of this weighting function are calculated according to

$$\begin{aligned} d(x_s - x_i) &= (x_s - x_{i+1})/(x_i - x_{i+1}) && \text{if } x_i < x_s\,, \\ d(x_s - x_i) &= (x_s - x_{i-1})/(x_i - x_{i-1}) && \text{if } x_i > x_s\,, \\ d(x_s - x_i) &= 1 && \text{if } x_i = x_s\,. \end{aligned} \tag{6}$$

Finally, these actual boundary values are compared with the desired boundary values ($\mathbf{V}$ and T_w) and the forcing terms are calculated using

$$\begin{aligned} \mathbf{f}_{\mathbf{m}(i,j)} &= (\mathbf{u}\cdot\nabla\mathbf{u})_{i,j} + \nabla P_{i,j} - \nu(\nabla^2\mathbf{u})_{i,j} \\ &\quad + \frac{1}{\delta t\, N_b} \sum_{s\in N(i,j)} D_{i,j}(\mathbf{x}_s)[\mathbf{V} - \mathbf{U}(\mathbf{x}_s)], \end{aligned} \tag{7}$$

$$f_{e(i,j)} = (\mathbf{u}\cdot\nabla T)_{i,j} - \frac{\nu}{\mathrm{Pr}}(\nabla^2 T)_{i,j} + \frac{1}{\delta t\, N_b}\sum_{s\in N(i,j)} D_{i,j}(\mathbf{x}_s)[T_w - T(\mathbf{x}_s)], \tag{8}$$

where $N(i,j)$ is the set of Immersed Boundary nodes associated with the current grid node and N_b is the total number of nodes in the set. These two equations use the same weighting functions as (5). In the case of Dirichlet boundary conditions the values of $\mathbf{V}$ and T_w are just set. For a Neumann boundary condition an extra layer of virtual nodes is created a small distance inside the immersed boundary nodes so that each node on the boundary has a corresponding node in the flow. The value of T_w is then calculated at each time-step using the extra nodes in the flow in order to give the desired temperature gradient at the boundary.

3 Results

3.1 Immersed Boundary Method accuracy

Simulations were performed to determine the accuracy of the Immersed Boundary. The vertical velocity very close to the hot wall compared across different grid sizes and the L_2 norms calculated are shown in Figure 3. This log-log plot shows a -2 slope for the L_2 norm. This slope indicates the Immersed Boundary maintains second order accuracy in space. This order is possible to achieve even though forcing is applied only to a single layer of cells because a bi-linear interpolation method is used and also because the number of Immersed Boundary points are increased independently of the grid.

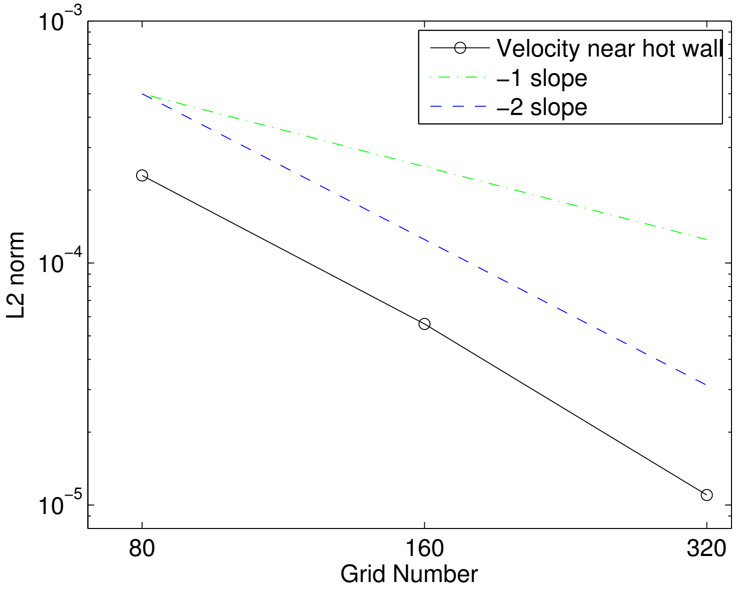

Figure 3: L_2 norm based on vertical velocity near the hot wall versus grid resolution

3.2 Velocity and temperature profiles

The following plots show velocity and temperature profiles in the standard cavity where the grid is aligned with the boundaries (no Immersed Boundary). On the same plots are results for the same cavity with the grid rotated relative to the boundaries, as described earlier. Therefore we expect to see the profiles look the same if the Immersed Boundary is not influencing the flow in some way.

All results use constant temperature and non-slip conditions ($u = v = 0$) for the side walls. The top and bottom walls have zero heat flux and are also non-slip. The critical parameters are the Rayleigh number (Ra) and the

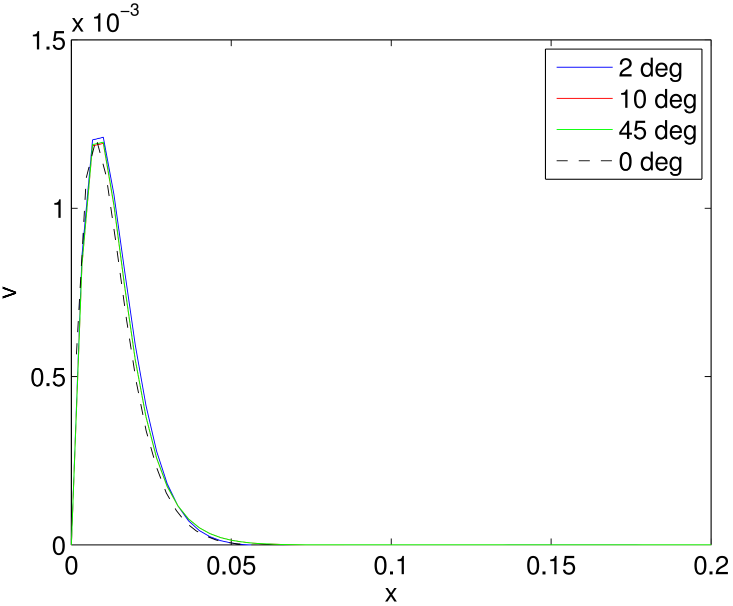

Figure 4: Velocity at cavity mid-height close to the hot wall. The full domain width is $x = 1.0$, hot wall temperature $300\,\mathrm{K}$, cold wall temperature $296\,\mathrm{K}$, $\mathrm{Ra} = 10^6$ and $\mathrm{Pr} = 7.0$.

Prandtl number (Pr) which are defined as

$$\mathrm{Ra} = \frac{g\beta\Delta TH^3}{\nu k} \quad \text{and} \quad \mathrm{Pr} = \frac{\nu}{k}\,, \tag{9}$$

where g is the acceleration due to gravity, β is the coefficient of thermal expansion, ΔT is the temperature difference between the hot and cold walls $(T_h - T_c)$, H is the cavity height, ν is the kinematic viscosity and k is the thermal diffusivity. In all simulations values are chosen to give $\mathrm{Ra} = 10^6$ and $\mathrm{Pr} = 7.0$, which result in a non-turbulent flow.

Figures 4 and 5 show very good correlation. The lines in Figure 6 are also very close to each other; the only appreciable variations occurring in Figure 7. In this figure the velocity in the boundary layers compares very well; however,

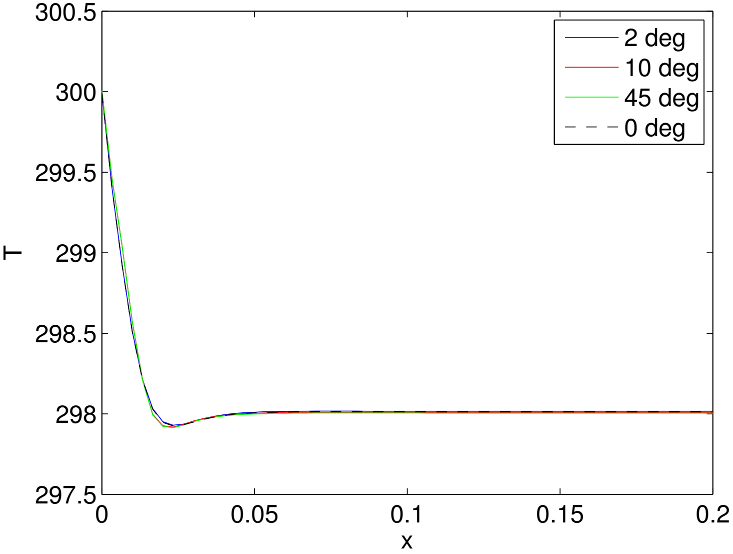

Figure 5: Temperature at cavity mid-height close to the hot wall. The full domain width is $x = 1.0$, hot wall temperature $300\,$K, cold wall temperature $296\,$K, $\mathrm{Ra} = 10^6$ and $\mathrm{Pr} = 7.0$.

it is the region where the boundary layer interacts with the central region that shows some discrepancy. Unfortunately the cause of this discrepancy is not clear at this stage.

4 Discussion

In general, the IBM cases correlate quite well with the base numerical simulation. The code has been validated previously in other similar studies, so it is therefore safe to use as the baseline.

Flow of this type generally involves a significantly finer grid in the vicinity of the boundary. One drawback of the IBM is that refining the grid in this

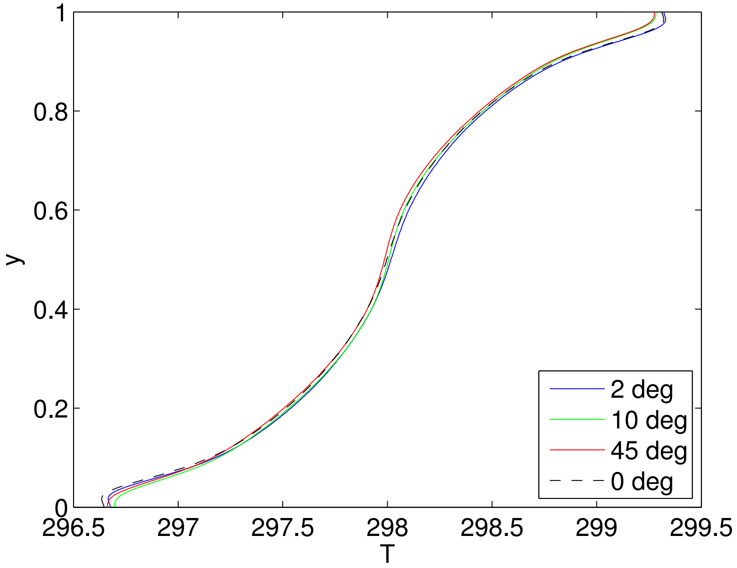

Figure 6: Temperature at cavity centre across full domain height. The hot wall temperature is 300 K, cold wall temperature 296 K, $\mathrm{Ra} = 10^6$ and $\mathrm{Pr} = 7.0$.

way is generally not possible. Therefore comparing the results to experiments will show poor results in the boundary layers. However, what these results intend to show is that the IBM can produce results similar to that for other equivalent numerical procedures.

5 Conclusions

An Immersed Boundary Method was successfully used to simulate a differentially heated cavity. The scheme maintains second order spatial accuracy and typical flow characteristics agree very well to a standard case. However, detailed results for turbulent flow have not yet been completed.

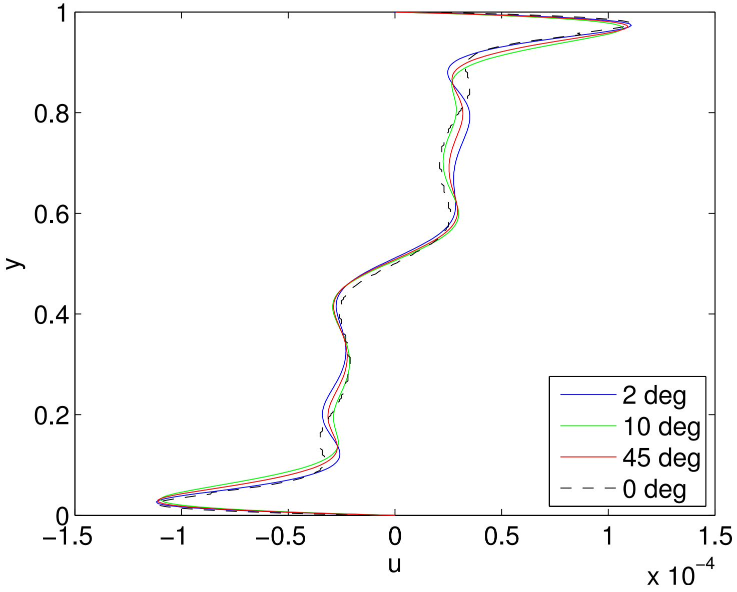

Figure 7: Velocity at cavity centre across full domain height. The hot wall temperature is 300 K, cold wall temperature 296 K, Ra $= 10^6$ and Pr $= 7.0$.

References

[1] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000. doi:10.1006/jcph.2000.6484 C537

[2] T. Gao, Y. H. Tseng, and X. Y. Lu. An improved hybrid cartesian/immersed boundary method for fluid-solid flows. *International Journal for Numerical Methods in Fluids*, 55(12):1189–1211, 2007. doi:10.1002/fld.1522 C537

[3] M. P. Kirkpatrick and S. W. Armfield. Experimental and large eddy simulation results for the purging of salt water from a cavity by an

overflow of fresh water. *International Journal of Heat and Mass Transfer*, 48(2):341–359, 2005. doi:10.1016/j.ijheatmasstransfer.2004.08.016 C538

[4] M. P. Kirkpatrick, S. W. Armfield, and J. H. Kent. A representation of curved boundaries for the solution of the navier-stokes equations on a staggered three-dimensional cartesian grid. *Journal of Computational Physics*, 184(1):1–36, 2003. doi:10.1016/S0021-9991(02)00013-X C538

[5] B. P. Leonard and Simin Mokhtari. Beyond first-order upwinding: The ultra-sharp alternative for non-oscillatory steady-state simulation of convection. *International Journal for Numerical Methods in Engineering*, 30(4):729–766, 1990. doi:10.1002/nme.1620300412 C538

[6] S. H. Peng and L. Davidson. Large eddy simulation for turbulent buoyant flow in a confined cavity. volume 22, pages 323–331, 2001. doi:10.1016/S0142-727X(01)00095-9 C537

[7] C. S. Peskin. Numerical-analysis of blood-flow in heart. *Journal of Computational Physics*, 25(3):220–252, 1977. doi:10.1016/0021-9991(77)90100-0 C536

[8] J. Salat, S. Xin, P. Joubert, A. Sergent, F. Penot, and P. Le Quere. Experimental and numerical investigation of turbulent natural convection in a large air-filled cavity. *International Journal of Heat and Fluid Flow*, 25:824–832, 2004. doi:10.1016/j.ijheatfluidflow.2004.04.003 C537

[9] Y. H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003. doi:10.1016/j/jcp.2003.07.024 C537

[10] N. Zhang and Z. C. Zheng. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics*, 221(1):250–268, 2007. doi:10.1016/j.jcp.2006.06.012 C538

[11] N. Zhang, Z. C. Zheng, and S. Eckels. Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method. *International Journal of Heat and Fluid Flow*, 29(6):1558–1566, 2008. doi:10.1016/j.ijheatfluidflow.2008.08.009 C537, C538

Author addresses

1. **K. A. Dittko**, School of Aerospace, Mechanical & Mechatronic Engineering, J07, University of Sydney, NSW 2006, Australia.
   `mailto:karl.dittko@sydney.edu.au`
2. **M. P. Kirkpatrick**, School of Aerospace, Mechanical & Mechatronic Engineering, J07, University of Sydney, NSW 2006, Australia.
3. **S. W. Armfield**, School of Aerospace, Mechanical & Mechatronic Engineering, J07, University of Sydney, NSW 2006, Australia.