Using the PCBCAT to Model Convective Heat Transfer from Electronic Devices on Printed Circuit Boards

Gerald W. Recktenwald
Department of Mechanical Engineering
Portland State University
Portland, Oregon

ABSTRACT

The Printed Circuit Board Convection Analysis Tools (PCBCAT) are computer programs for predicting the thermal performance of convectively cooled printed circuit boards. The flow field is obtained by solving the depth-averaged (DA) momentum and continuity equations for the fluid layer above the electronic components. The two-dimensional DA velocity field data is then incorporated into a solution for the three-dimensional temperature field in the fluid and electronic components. By solving the conjugate heat transfer problem the need to specify a heat transfer coefficient is eliminated. By solving the DA flow equations instead of the three-dimensional Navier-Stokes equations the computing time is reduced substantially. The physical problem to be analyzed is specified via commands in a plain text file which is parsed by the PCBCAT preprocessor. The commands allow the user to describe features of the calculation domain in a succinct, physically intuitive way. This paper provides an overview of the PCBCAT package from the user's perspective. The modeling capabilities of the PCBCAT are demonstrated with two example problems.

Introduction

The computer codes described in this paper began with research into the feasibility of using depth-averaged (DA) modeling to simulate the flow over printed circuit boards (Recktenwald & Butler, 1991). It soon became apparent that although the flow field could be simplified with DA modeling, the temperature field needed to be solved in three dimensions in order to avoid specifying heat transfer coefficients. A code to solve the three-dimensional energy equation was written and used to simulate a number of heat transfer experiments that model convectively cooled electronic components (Ma & Recktenwald, 1993). The experiments involved uniform arrays of heated blocks in large aspect ratio (width to height) ducts. It was possible to get reasonable results for these simulations by assuming the flow was unidirectional, so solving the DA flow field was not necessary. The next step was to combine the DA flow model with the three-dimensional energy equation code as documented in a companion paper (Recktenwald, 1995b).

In addition to developing the basic solution procedure outlined above, a substantial effort has gone into writing the analysis codes so that they can solve any problem involving flow over printed circuit boards in confined spaces. The result of this effort is the Printed Circuit Board Convection Analysis Tools, or PCBCAT. The PCBCAT are actually a collection of programs that work together, although the user can think of these programs as a single entity.

The PCBCAT consist of a preprocessor, the depth-averaged flow model, and the energy equation model. The preprocessor is the user interface to the analysis codes. It reads a plain text file that contains commands and arguments that describe a circuit board model. Though this text-based user interface is somewhat old fashioned by the standards of modern computer packages, the preprocessor commands provide a simple and relatively free form language with which a user describes the physical features, prescribes the flow type, and controls the execution of the analysis programs. The PCBCAT manual (Recktenwald, 1995a) contains a complete description of how to build and analyze circuit board models with the PCBCAT.

The flow field over printed circuit boards is computed by solving a control volume finite difference approximation to the depth-averaged momentum and mass conservation equations. Throughout the remainder of the paper this will be referred to as the DA model. The DA flow field is then used in the three-dimensional energy equation code to predict the temperature field in the fluid and within the heat-generating electronic devices. In other words the energy equation model solves the three-dimensional conjugate heat transfer problem using the flow field supplied by the DA model.

The PCBCAT is not meant to compete with fully 3D CFD codes or commercial circuit board analysis packages. Rather, it is an attempt to share the results of our research, and we hope, to stimulate further developments in circuit board thermal modeling. Though the PCBCAT codes are still under development, the package is at a point where it can be used for public experimentation and testing. The latest version of the PCBCAT, including a manual, is available free for noncommercial use. Directions for downloading the programs and documentation are given in the Appendix to this paper.

The primary purpose of this paper is to introduce the PCBCAT package and to give an overview of its capabilities from a user's perspective. The following sections contain descriptions of how a given physical problem is setup and solved, what output is provided, how that output can be incorporated into vizualization software packages. Finally, representative results of using the program are presented.

Flow of Information and Control

The PCBCAT programs execute in sequence and communicate via data files. The flow of information between these programs is depicted in Figure 1. A model of a particular board is completely described by a text file that is represented in Figure 1 by the box labeled "user.input". (The preprocessor input file can have any name, user.input is a generic name taken for the purpose of exposition.) The preprocessor reads user.input and creates a temporary file labeled "pcbcat.cntl". The pcbcat.cntl file is a translation of the user's description of the problem into a format that is easily readable by the analysis codes. Whereas the data in user.input is relatively free form and contains comment statements, the pcbcat.cntl file has a rigid structure and no extraneous text. The pcbcat.cntl file also contains details about the grid, boundary conditions, and electronic components that are implied by the data supplied in user.input. This is described in more detail in the "preprocessor" section, below.

The pcbcat.cntl file controls the execution of both the DA and 3D energy equation models. These analysis codes create their own output in the form of plain text files
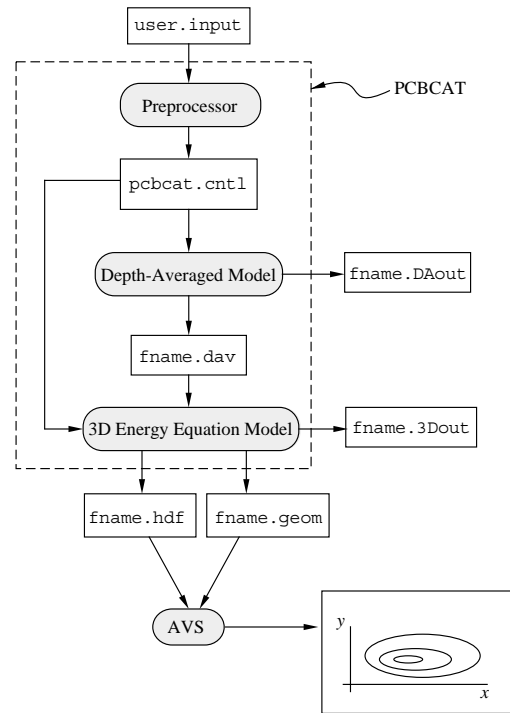

Figure 1: Schematic of the flow of data between the programs. Executable programs are depicted as shaded oval boxes. Data files input to or output from the programs are shown as rectangular boxes. The box in the lower right corner represents the graphical output from AVS, in this case a contour plot.

and binary files. Plain text files are meant to be read by the user. They contain summary information useful for further engineering analysis, such as calculation of heat transfer coefficients. Binary files contain field data meant either to be read by the 3D energy equation model or by visualization software.

A preprocessor command allows the user to specify a base name with which the PCBCAT will construct names of the output files. In Figure 1 the base name is the generic string, "fname". The DA model creates the `fname.dav` and `fname.DAout` files. The `fname.DAout` file is a text-only summary of the analysis. The `fname.dav` file contains the depth-averaged velocity field used as input to the 3D energy equation model.

The 3D energy equation model creates three files, `fname.3Dout`, `fname.hdf` and `fname.geom`. The `fname.3Dout` file is a text-only summary of the analysis. The `fname.hdf` and `fname.geom` files contain the field and object-based data designed to be imported into visualization programs.

The temperature field calculated by the 3D energy equation model is stored in `fname.hdf` in Hierarchical Data Format (HDF). This format was developed at the University of Illinois. It is in the public domain, and many visualization packages can import HDF files. We have written interactive modules for the AVS visualization system (AVS Inc., 1992) that allow the user to import the HDF output from PCBCAT along with the problem description data saved in `fname.geom` (Recktenwald & Gotseff, 1995). With this system users can render a three-dimensional model of the circuit board, and using mouse input, interactively query the objects in the domain.

The PCBCAT Preprocessor

Early versions of the DA and 3D energy equation codes had no user interface. Each new physical problem required a custom code. This was time consuming, especially since the debugging and testing process needed to be repeated. The initial objective in developing the preprocessor was to eliminate the need to rewrite the codes for each new physical problem to be analyzed. This required that the analysis codes be structured to accept a variety of boundary conditions, flow orientations, and flow regimes. Such flexibility demands a substantial amount of input data in order to completely specify all of the free parameters that determine the flow. The data needed just to define the computational grid, for example, can become unwieldy for a problem with complex geometric details. Large input data sets do allow flexibility, but also are a liability since it can make the analysis tool difficult to use, and it can increase the likelihood of data entry errors.

In recognition of the tension between flexibility and complexity, a fundamental design criteria for the preprocessor was established. Any feature or option requested by the user should be specified in the most compact way possible, while at the same time giving user complete control. This lead to the development of what we call the object-based problem description. In this context an object is any physical entity with geometric, thermophysical, hydrodynamic or thermal properties that affect the solution. (This use of the word object is not to be confused with object-oriented programming or object oriented software design. The current version of the PCBCAT is written in ANSI C using procedural algorithms and conventional data structures.)

The following additional criteria were used in the preprocessor development: the preprocessor should be extensible; it should contain error checking, and when an error is encountered the code should print a meaningful error message and stop in a controlled way; it should allow relatively freeform input; and it should be capable of being modified or even replaced with minimal impact on the analysis codes. These criteria lead to a substantial software development effort resulting in the current version of the PCBCAT.

The user controls the PCBCAT through a plain text file consisting of preprocessor commands and their arguments. The commands either set flags, set control parameters, or specify objects. Flag setting commands have no arguments, and merely serve to turn an option on or off. An example is the `INTERACTIVE` command that instruct the analysis codes to prompt the user for options during execution. The default is to complete the analysis without any prompting.

Commands that set control parameters have between one and three arguments. These commands either control the numerical solution process or apply some condition to the entire calculation domain. Two examples are the `DOMAIN` command, which has three numerical arguments to specify the overall dimensions of the calculation domain, and the `COOLANT` command which has one argument to specify fluid flowing through the domain. The argument of the `COOLANT` command is a text string such as "water" or "air" that refers to entries in a user-editable database of thermophysical properties. Providing material properties in a separate database gives the user complete control, while minimizing the risk of data entry errors. It also makes the preprocessor input file easy to read and modify. The size of the material database is unlimited.

Another category of preprocessor commands is used to specify objects in the domain. These objects are helpful abstractions for user input and for the internal logic of the DA and 3D energy equation codes. There are three fundamental object types—probes, patches and blocks—and each object type has its own preprocessor command. The user can specify an unlimited number of each type of object. As the number of objects increases, the model becomes more complex and the execution time of the analysis codes will also increase.

Probes are used to selectively sample the results of the calculation. At a probe location the temperature and velocity components are reported.

Patches are generic, two-dimensional objects that may be attached to any bounding surface of the domain. The PATCH command is used to assign hydrodynamic and thermal boundary conditions over regions that do not completely cover one of the six bounding surfaces of the domain. Inlets and Outlets are special types of patch objects, and these have their own commands, INLET and OUTLET.

Blocks are three-dimensional solid objects. These are used to represent internal obstacles, or the simple heated blocks used in may experiments that simulate electronic cooling situations. The BLOCK command locates a block, specifies its size, identifies its material, and sets its volumetric heat generation rate.

A fourth type of object called a "device" is under development. It will be used to model electronic components. The DEVICE command requires a user to specify the name of the device, its orientation with respect to the coordinate system of the board, its location on the board, and its relative heat generation rate (percent of rated power). The remaining information, including the geometry of each device, is defined in another user-editable database.

All objects are located in the calculation domain without direct reference to the grid system used in the analysis. One of the major tasks performed by the preprocessor is the creation of a grid that fits all of the objects. The preprocessor actually constructs a very coarse grid—the object grid—that is just sufficient to locate all edges of the geometric objects specified by the user. The user sets tolerances on the largest acceptable size of the control volumes and the analysis codes (not the preprocessor) generate fine grids by subdividing the object grid until the user-specified tolerances are met. The analysis grid is consistent with the object grid in that the control volume faces are aligned with the edges of the objects in the domain. After the analysis grid is created, the analysis codes use the preprocessor-defined objects to specify material properties, boundary conditions, and ultimately, to extract data for presentation to the user as output.

There is no limit on size of the grid or the number of objects in a given problem so long as the computer running the model has sufficient memory.

A Simple Example

To demonstrate how the PCBCAT works consider the simple example of flow past a single, heat-generating block mounted to the bottom of a duct. This situation is depicted by the sketch in Figure 2. Air enters the domain through the y-z plane at $x = 0$ and exits through the y-z plane at $x = 30$ cm. No-slip, adiabatic boundaries are applied on the remaining four walls. Although this model problem is far from a real circuit board, it provides a succinct introduc-
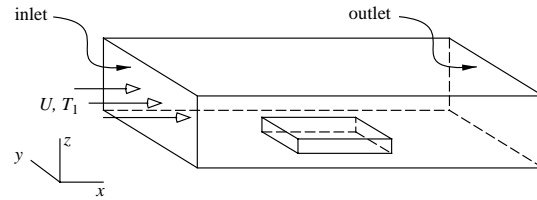

Figure 2: Model problem of flow past a single heated block.

```
#  PCBCAT example problem "block1"
#  Flow past a single heated block.

#  Set the base name for output file names
FILE_NAME    block1

#  Define a 30x20x4 cm computational domain
DOMAIN  0.30  0.20   0.04

#  Set tolerance on control volume sizes
CV_SIZE 0.01  0.01   0.002

#  Global parameters
COOLANT         air
ITER_CONTROL  medium     50
FLOW_REGIME    turbulent
FLOW_FIELD     depthAve
FLOW_PROFILE   fully_dev

# Inlet and outlet cover east and west boundaries
INLET   west 0.0 0.20 0.0 0.04 10.0 1.131 0.0 0.0
OUTLET east 0.0 0.20 0.0 0.04

#  Remaining boundaries are adiabatic walls
BOUND   north   fluxBC   0.0    wall
BOUND   south   fluxBC   0.0    wall
BOUND   up      fluxBC   0.0    wall
BOUND   down    fluxBC   0.0    wall

# heated 5x5x1 cm Al block in the center of the duct
BLOCK heater aluminum 2 1 0.125 0.05 0.075 0.05 0.01

#  Probe locations are scaled to domain dimensions
PROBE   relative  0.50   0.5  0.125
PROBE   relative  0.75   0.5  0.5
```

Figure 3: PCBCAT input file for cooling of a single block in a channel.

tion to using the PCBCAT. More realistic problems differ only in the number of preprocessor commands necessary to specify all of the objects in the domain. The modeling strategy is the same.

Figure 3 is the input file for the PCBCAT preprocessor to completely set up and control the analysis of the situation represented in Figure 2. The file in Figure 3 contains comment statements and preprocessor keywords. The comment statements begin with the "#" character in the first column. The keywords are in all capital letters. The first keyword, FILE_NAME, specifies a text string, "block1", as the base name from which the PCBCAT will construct output file names. The DOMAIN keyword sets the extent of the overall calculation domain. It has three arguments which are floating point values specifying the x, y and z dimensions of the

domain in meters. The `CV_SIZE` keyword sets the maximum allowable size of the control volumes in each coordinate direction. These tolerances, and the geometric data for objects in the domain are all that is needed to specify the computational grid.

The next group of preprocessor commands set parameters that affect the overall model. The `COOLANT` command tells the preprocessor that the fluid in the domain has the thermophysical properties of air, as defined in the material database. The `ITER_CONTROL` command sets the convergence tolerance and the maximum number of iterations for the DA and 3D models. The "medium" convergence tolerance defines several numerical tolerances on residuals of the linear equations that are solved in the analysis programs (Recktenwald, 1995a). The `FLOW_REGIME` command specifies whether or not a turbulence model is to be used. When the argument is "laminar" the effective viscosity and diffusivity are equal to their molecular values. When the argument is "turbulent", the turbulence model described in (Recktenwald, 1995b) is used.

The `FLOW_FIELD` command determines whether a uniform flow field or the depth-averaged flow field is to be used in solution to the 3D energy equation. A uniform flow field is not meaningful in all cases. In this simple example the results of solving the energy equation with a uniform flow field could be compared with the results of using the DA flow field. The "depth_ave" argument means that the DA flow field will be calculated. Finally, the `FLOW_PROFILE` command is used to select the z-direction variation of the velocities. Choosing "fully_dev" for the argument of the `FLOW_PROFILE` command means that the vertical profiles will be computed with the fully-developed profile consistent with the argument of the `FLOW_REGIME` command.

The `INLET` keyword has several parameters. The first is "west", a text string that specifies the face on which the inlet is located. All directions are designated with the compass point naming convention where the positive y direction is north, the positive x direction is east, and so on. The next four arguments of the `INLET` command give the location of the inlet on the west face. For this simple example the inlet covers the entire face. It's possible, as shown in the next example, that inlets and outlets cover only part of a domain face. The last four arguments of the `INLET` command assign the inlet temperature ($10\ ^\circ C$), and inlet velocity components, $(v_x, v_y, v_z) = (1.131, 0, 0)$.

Boundary conditions are prescribed with the `BOUND` command. The first argument specifies the face, the second and third arguments define the thermal boundary conditions, and the fourth argument prescribes the hydrodynamic boundary condition. Since the east and west boundaries are already defined as inlets and outlet, respectively, only the remaining four domain boundaries need to be treated. All four are adiabatic, as indicated by the "fluxBC 0.0" values

of the second and third arguments. The "wall" value of the fourth argument selects a solid boundary for which no-slip hydrodynamic boundary conditions will be applied.

The heated block is specified with the `BLOCK` command. The arguments specify the block material properties ("aluminum"), its total power dissipation, ("2.0" Watts), it's heating status ("1" indicating that the heater is on), and the location and size of the block. The heating status is redundant with the total power dissipation, in apparent violation of the principle that the user should have to specify the absolute minimum amount of information necessary. In this case the design principle was relaxed so that a user could turn off a heating element without having to reset its nominal power setting.

The last two lines of the input file define two probes via the `PROBE` command. The first argument of the `PROBE` command is either "relative" (as in Figure 3) or "absolute". A "relative" probe position means that the preprocessor will interpret the last three arguments as fractions of the x, y and z dimensions of the domain. For example the first probe in Figure 3 is located in the geometric center of the x-y plane and one eighth of the z-direction domain length from the bottom plane. The alternative is to locate the probe by the absolute coordinate position, in meters.

The single heated block problem shows how to perform thermal analysis with the PCBCAT. Though it helps to demonstrate features of the preprocessor it is not a particularly difficult or interesting problem to analyze. Using the problem description in Figure 3 the PCBCAT solves the flow field in 13 seconds and the temperature field in 22 seconds on a SUN SPARC 10/40 workstation. Figure 4 shows the temperature contours in an x-y plane at the base of the domain. The solution was obtained on a $n_x \times n_y \times n_z = 33 \times 23 \times 22$ grid, where n_x is the number of nodes in the x-direction, etc. The temperature field is symmetric, as expected and there is a wake downstream of the block.

Figure 5 is a plot of the pressure along the centerline of the duct. The discontinuities in pressure result from the discontinuous changes in fluid layer depth at the leading and trailing edges of the block. The spikes in the pressure field appear to be inherent in the DA model. We have performed independent calculations with alternative formulations of the pressure terms in the momentum equations and the pressure spikes remain.

A Computer Motherboard

This section presents results of applying the PCBCAT to a fictitious computer motherboard mounted in a cabinet. Figure 6 contains a plan view of the motherboard, along with front and rear views of the cabinet. On the front of the cabinet is a long inlet slot. On the rear of the cabinet is a fan. Since the PCBCAT is based on a Cartesian grid it is not possible to represent a round opening so the fan is modeled as
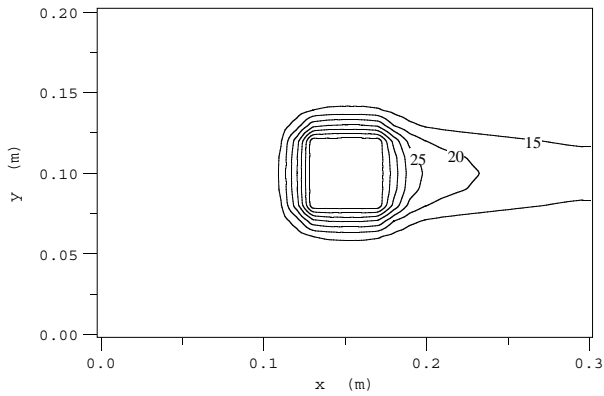
Figure 4: Temperature field in an x-z plane at the base of the domain for the model problem of flow past a single block. Temperature contours are at $5\,^{\circ}C$ intervals. Maximum temperature in the center of the block is $48.6\,^{\circ}C$.
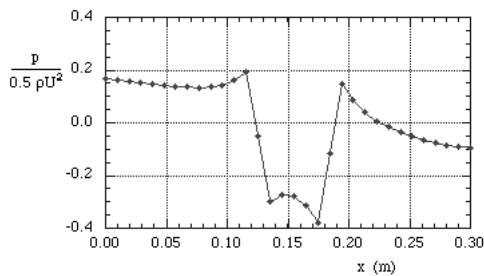

Figure 5: Normalized pressure distribution along the centerline of the channel for the model problem of flow past a single block.

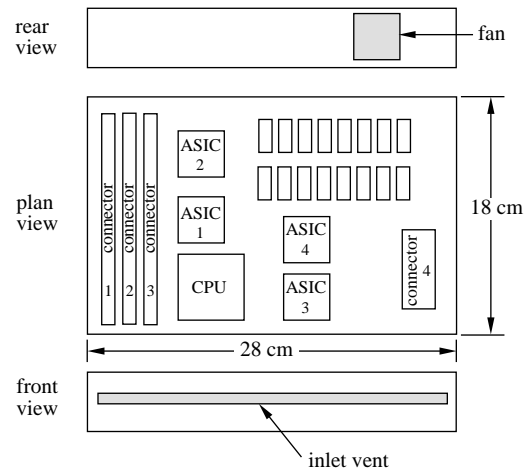Figure 6: Example problem of a motherboard in a computer cabinet.

a square outlet.

The motherboard model is an idealized representation of a computer. There is no power supply or hard drive and there are no expansion cards projecting from the motherboard. Although it is a simplified view of reality this example demonstrates the capabilities of the current version of the PCBCAT.

The motherboard is populated with a CPU, four custom IC packages (ASIC), three card edge connectors, and two banks of RAM chips. Each of the electronic components (CPU, ASICs, RAM) are modeled as solid blocks of epoxy with uniform heat generation. In the near future the "device" data type mentioned in the preceding sections will be capable of greater realism in component modeling.

Figure 7 shows the depth-averaged stream-lines obtained on a 57×34 grid. The flow turns from the inlet slot toward the fan, and it develops a recirculation zone in the upper left corner.

Figure 8 is a surface plot of the depth-averaged pressure field for the flow field depicted in Figure 7. The largest pressure change occurs at the outlet, where the fluid is accelerated before leaving the domain. Note that the pressure surface in Figure 8 has been rotated so that the outlet is in the vertical plane closest to the observer. The location of the outlet is apparent by from the large depression in the pressure surface. There are small anomalies in the pressure field in the center of the domain, but the large spikes evident in Figure 5 are absent.

Figure 9 shows the temperature contours on a plane 0.19 cm from the bottom of the domain. The temperature field was obtained on a $57 \times 34 \times 37$ grid. The thermal wake of the CPU is clearly visible.
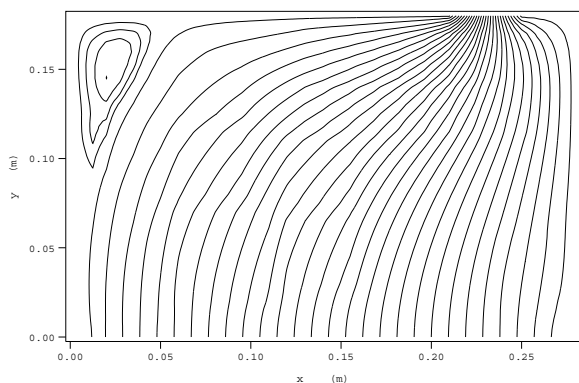
Figure 7: Depth-averaged stream function contours for the motherboard model on a 57 by 34 grid.
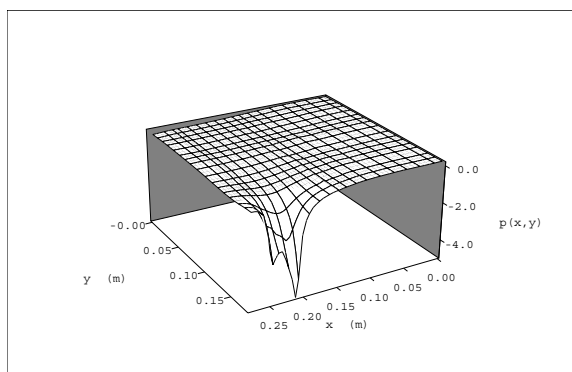

Figure 8: Surface plot of the depth-averaged pressure field for the motherboard model. The pressure field is computed on a 57×34 grid.
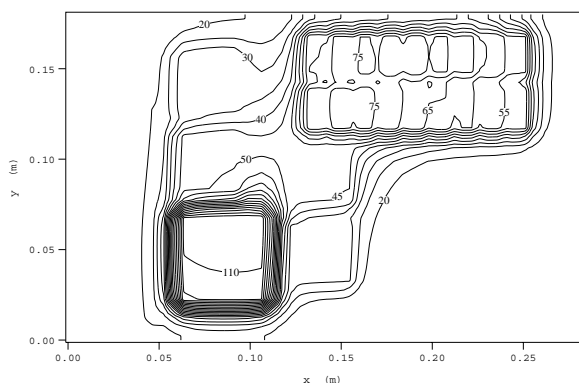

Figure 9: Temperature contours in a plane 0.19 cm from the bottom of the domain. The temperature field is computed on a $57 \times 34 \times 37$ grid.

$n_x \times n_y \times n_z$	DA execution time (seconds)	3D execution time (seconds)
$37 \times 20 \times 12$	9	21
$50 \times 28 \times 30$	30	228
$57 \times 34 \times 37$	81	693

Table 1: Execution times for motherboard model problem run on a Sun SPARC 10/40. For the DA model only the n_x and n_y grid sizes are relevant.

Execution Time

Table 1 gives some representative execution times for the motherboard model problem on a progression of finer grids. The timing results were obtained on a SUN SPARC 10/40, an entry level unix workstation roughly two years old. High end personal computers with the latest generation CPUs are approximately as fast. The longest run-time in Table 1 is about eleven and a half minutes, considerably less than that required for a fully 3D CFD analysis.

Conclusion

The PCBCAT embody a novel approach to the analysis of convectively cooled printed circuit boards. The flow field above the board is computed with a depth-averaged model, and the temperature distribution in the fluid and solid components is obtained by solving the three-dimensional energy equation. The depth-averaged flow model allows complex problems to be analyzed in a small fraction of the time needed to obtain comparable solutions with a full three-dimensional CFD code.

PCBCAT models are constructed with simple commands in a text-only input file. Results are reported in text files and in binary files suitable for import to popular visualization packages. The PCBCAT codes are an experiment in the development of design tools for packaging engineers.

Although they do not have a flashy user interface, the PCBCAT contain the essential analysis and data manipulation tools necessary for a complete thermal simulation of convectively cooled printed circuit boards. Enhancements and ports of the PCBCAT to other platforms are underway. Users are encouraged to download and experiment with the current version.

Appendix: Downloading the PCBCAT

As of this writing the PCBCAT codes are only available for SUN workstations. Ports to other platforms are in progress. Contact the author (`gerry@me.pdx.edu`) for updated information, and check the ftp site and the PCBCAT home page for new versions.

The latest version of the PCBCAT and the PCBCAT manual can be obtained by anonymous ftp with the steps given below, or by pointing a web browser to `http://www.me.pdx.edu/~gerry/PCBCAT/` and following the links to the "Download" page.

If you do not have a web browser you can use anonymous ftp directly. From any computer that is on the Internet and that supports ftp, perform the following commands

```
ftp ee.pdx.edu
    (login with username ''anonymous''
     and your e-mail address as password)
cd pub/users/faculty/gerry/PCBCAT
mget *
quit
```

Additional instructions are in the `ReadMe` file on the ftp server.

REFERENCES

AVS Inc. 1992. *AVS User's Guide*. Advanced Visual Systems, Inc., Waltham, MA.

Ma, Yidong, & Recktenwald, Gerald W. 1993. Simplified Simulation of Convective Heat Transfer From an Array of Heated Blocks in a Rectangular Channel. *Pages 609–618 of:* Engel, Peter A., & Chen, William T. (eds), *Advances in Electronic Packaging 1993*, vol. EEP-Vol. 4–2. ASME.

Recktenwald, Gerald W. 1995a. *The PCBCAT Manual*. Portland State University, Mechanical Engineering Department, Portland, OR.

Recktenwald, Gerald W. 1995b. Prediction of device temperatures with depth-averaged models of the flow field over printed circuit boards. *In: Proceedings of 1995 International Mechanical Engineering Congress and Exposition*. San Francisco, CA: ASME.

Recktenwald, Gerald W., & Butler, Peter O. 1991. Depth-averaged modeling of convective heat transfer from printed circuit boards. *In: Proceedings of ASME Winter Annual Meeting*, vol. 91-WA-EEP-36. Atlanta, GA: ASME.

Recktenwald, Gerald W., & Gotseff, Peter A. 1995. A Visualization Tool for CFD Models of Convectively Cooled Printed Circuit Boards. *In: Proceedings of 1995 ASME National Heat Transfer Conference*. Portland, OR: ASME.