

Customizable Multidiscipline Environments for Heat Transfer and Fluid Flow Modeling

B. Cullimore, S. G. Ring, J. Baumann
C&R Technologies, Inc.

ABSTRACT

Thankfully, the age of stand-alone fixed-input simulation tools is fading away in favor of more flexible and integrated solutions. "Concurrent engineering" once meant automating data translations between monolithic codes, but sophisticated users have demanded more native integration and more automated tools for *designing*, and not just *evaluating* point designs. Improvements in both interprocess communications technology and numerical solutions have gone a long way towards meeting those demands.

This paper describes a small slice of a larger on-going effort to satisfy current and future demands for integrated multidisciplinary tools that can be highly customized by end-users or by third parties. Specifically, the ability to integrate fully featured thermal/fluid simulations into Microsoft's Excel™ and other software is detailed. Users are now able not only to prepare custom user interfaces, they can use these codes as portals that allow integration activities at a larger scale. Previous enabling technologies are first described, then examples and repercussions of current capabilities are presented, and finally in-progress and future technologies are listed.

BACKGROUND AND HISTORY

Almost all mechanical engineering simulation tools, including the ancestral SINDA and TRASYS programs and many of their descendents, began as batch-style stand-alone analyzers with predefined solution sequences (e.g., given fluxes and conductances, predict temperatures).

The advent of graphical user interfaces (GUIs) in the 80's and 90's helped tremendously by reducing the effort required to build and verify thermal models. Similarly, automated translation and import/export tools between CAD and FEM-based structural codes, epitomized by codes such as Thermal Desktop®* (Ref 1), have eliminated arduous and error-prone tasks. Such interface improvements, automated conversions, and streamlined communications continue to be introduced every year for new codes and even new classes of codes.

However, such conversions and translations are usually restricted to "one-time" events during the process of build-

ing a model. For example, CAD geometry can be imported and either converted into a thermal model or can be used as scaffolding to "snap on" thermal modeling objects, but changes such as altered design dimensions often must be re-propagated manually.

A big change began to occur around 1990 with the introduction and increased acceptance of parametric CAD modeling tools such as Parametric's ProEngineer®. Dimensions and other attributes could be described parametrically rather than as fixed numeric values. Today, all major mechanical design packages feature parametric modeling: rules, variables, and algebraic expressions by which designs can change and adjust themselves.

Unfortunately, most mechanical design *evaluation* packages (i.e., simulation tools) have missed the revolution. A few exceptions exist such as ANSYS®, and many other structural codes are catching up. Thermal analysis tools, as a class, have tended to lag behind.

SINDA/FLUINT was rewritten internally in the mid 90's to take advantage of parametric modeling (Ref 2). Models could be completely rescaled to new geometries and properties *while solutions were in progress*. Thermal Desktop and its associated modules RadCAD® and FloCAD® were created in the last 10 years, and therefore their design could take advantage of foresight instead of requiring retrofits: by that time the popularity of parametric modeling methods was well established.

This popularity arose from increased ease-of-use, such as the ability to make sweeping changes to a thermal model in a simple, centralized fashion. Changes to a single dimension (say, diameter or length) are automatically propagated through the model, even while it is being solved. This naturally lead to automation not only of *changes*, but of *tasks*: SINDA/FLUINT could be used to *select* one or more values perhaps to size or optimize them (Ref 3) or to automatically calibrate them against test data (Ref 4), or to perform statistical design studies (Ref 5).

The addition of parametric modeling is therefore well worth the programmer's effort if for no other reason than the enhancement it provides to stand-alone codes. But even more power can be gained in a multidisciplinary world via the integration of multiple tools, especially if each such tool is capable of parametric modeling. Instead of (or in addition to) conversions or translations of sets of data, groups of

* Thermal Desktop®, RadCAD®, FloCAD® are registered trademarks of C&R Technologies®. Other trademarks are owned by their respective companies.

otherwise independent models could communicate just by passing a few simple values to each other.

The ability to command wholesale changes to a model from outside of that code via the passing of a few parameters not only enables multidisciplinary design environments, it allows users to write their own GUIs and their own custom communication pathways, as will be detailed in this paper.

PRIOR CONNECTIONS

This section provides examples of interprocess communications roughly in the historical order of their creation. This format provides a logical sequence for the discussion of various developmental issues: how problems arose and how they were resolved.

ISIGHT® CONNECTION

The ability to make external yet fully automated changes to a executing thermal/fluid model was first performed for the OptiOpt™ project (Ref 6 and 7). OptiOpt was a pathfinder project aimed at identifying and removing bottlenecks in multidisciplinary analysis of thermal, structural, and optical systems.

The integration framework iSIGHT® (from Engineous) was chosen, although many other similar frameworks exist and have matured since that original election. Significant performance enhancements can be exploited if Thermal Desktop and SINDA/FLUINT could remain “persistent,” which in iSIGHT terminology means that they would remain active and “listening” rather than restarted each iteration. Every time a new design was to be evaluated (i.e., a change in dimension, property, control system, etc.), iSIGHT design parameters are sent to Thermal Desktop as “symbols” and/or perhaps directly to SINDA/FLUINT as “registers,” commanding sweeping model changes.

Dynamic Thermal Desktop—A first step was to enable Thermal Desktop to launch a SINDA/FLUINT run and to establish communications with SINDA/FLUINT such that user logic could be used to command refreshed Thermal Desktop radiation or contact/conduction calculations. SINDA/FLUINT could then accept the new network information while still executing (i.e., without stopping, preprocessing, and recompiling). This “dynamic mode” is a mature capability that has been described in prior papers, with the current status best summarized in Reference 1. This feature allows the two codes to automate important tasks such as thermal design optimization, calibration of thermal parameters (including optical properties) to test data, etc. Via dynamic reread of structural node points, certain thermoelastic analysis tasks can be automated as well. A second step was necessary, however, to open up this system to external drivers and therefore to provide a truly multidisciplinary environment (Figure 1): the acceptance of general Microsoft COM-based commands.

While detailed discussions of the workings of COM are beyond the scope of this paper, what COM does can be described simply as enabling applications to provide a run-

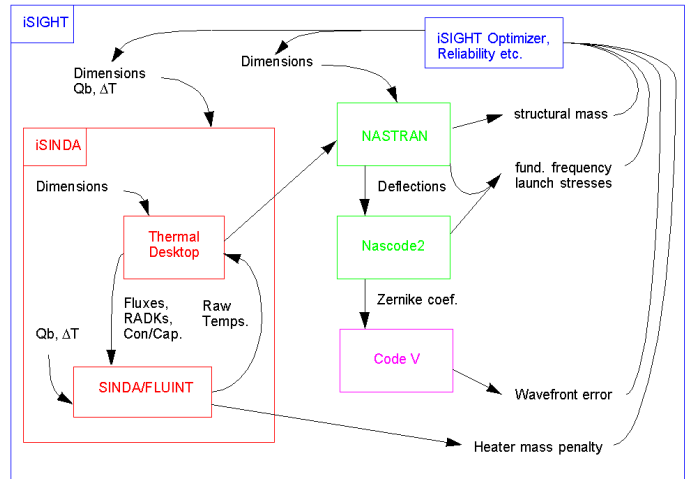


Figure 1: Thermal Desktop Dynamic Mode plus COM Connections to Engineous' iSIGHT and File Connections to MSC.NASTRAN

time callable interface to the outside world. In essence, executing applications can be made to look like callable subroutines, and the connections between these subroutines can be established dynamically at run-time. Instead of being linked together at compile-time, applications automatically find each other's interfaces and connect to them through operating system services provided by COM.

The creation of COM-aware SINDA, however, represents a special challenge. The SINDA main program does not exist until a run is made, yet COM is designed to connect to definite (pre-existing) executables. Many thermal programs like SINDA/FLUINT accept Fortran-based user logic, and therefore they perform compilations and then link with a library each time a run is made: the processor is “freshly made” each time SINDA-like codes are executed.

This special nature of SINDA does not represent a challenge when Thermal Desktop launches SINDA/FLUINT dynamically as described above, nor does it represent a problem for third-party codes which launch Thermal Desktop (which in turn might launch SINDA/FLUINT). However, for third-party codes such as iSIGHT (as well as MATLAB, Excel, and others to be described later) which launch SINDA/FLUINT directly, a new technique was required.

Generalized COM Methods—To resolve this problem, a new communication layer was required. Third-party programs such as iSIGHT launch a SINDA/FLUINT “controller” as a COM object (“iSINDA” in Figure 1). These programs tell the controller the name of the model to run, and the controller then takes care of launching the SINDA/FLUINT run.

Once launched, the user calls a special utility called COMINIT within their SINDA/FLUINT model (usually within OPERATIONS, the central driver). The COMINIT routine knows to find the running controller and connect back to it. Once this connection is established, the user can call various wait and data fetch/set routines inside SINDA/FLUINT. When waits are called, the controller returns to its client

(e.g. iSIGHT) so that the third-party program can set information in the SINDA/FLUINT run before control is returned to SINDA/FLUINT. The cycle of waiting, data exchange, and restart can continue indefinitely as needed to accomplish parametric sweeps, optimizations, statistical analysis, or perhaps as determined interactively by the user.

This communication technique (summarized in Figure 2) is common to the MATLAB and Excel connections described later, and can be utilized by any COM-aware software requiring a complex thermal/fluid companion solution.

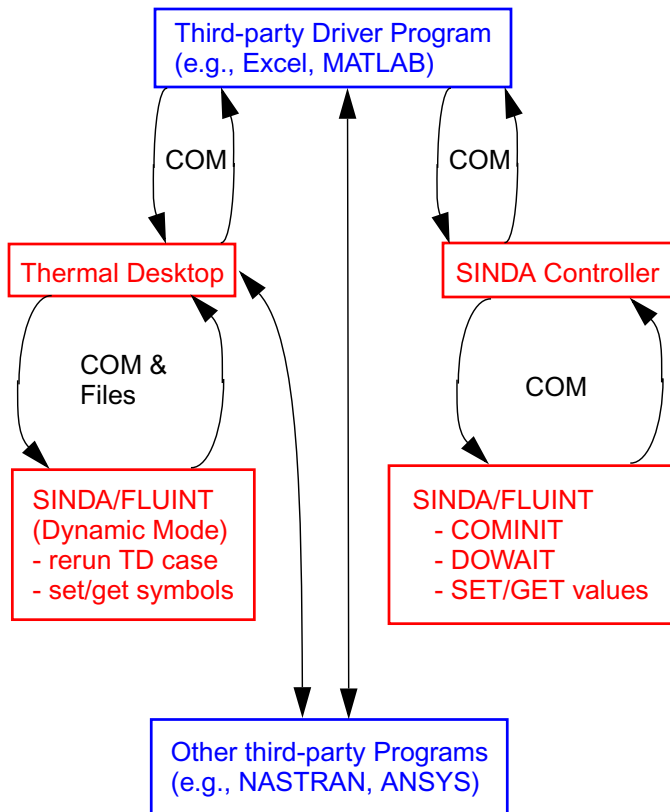


Figure 2: COM-based Access to Full-featured Thermal/fluid Companion Solutions

MATLAB® CONNECTION

Once thermal/fluid solutions were fully parametric and once COM-based methods were available for including arbitrarily complex thermal/fluid models, applications arose quickly to exploit the new capabilities.

Some of the first such applications involved MathWorks' MATLAB® (and, optionally, Simulink®).

NREL's Advisor—The US Department of Energy's National Renewable Energy Laboratory (NREL) created vehicle-level energy models to help guide efforts to meet gasoline mileage and emissions targets in next-generation cars and light trucks. Their MATLAB-based Advisor program required detailed transient simulations of air conditioner performance in order to predict compressor torques on the

engine during the standard drive cycles used to verify compliance with US regulations (Ref 8).

SINDA/FLUINT is one of the few thermohydraulic codes with the capability to predict vapor compression cycle in a transient mode, and a detailed model of such a cycle is necessary to predict current torque on the engine. Therefore, a SINDA/FLUINT model (consisting of one R134a fluid submodel and various thermal submodels representing evaporator and condenser structure) was invoked simultaneously with an Advisor solution. Every (approximately 1 second) time step, the Advisor code predicted current engine RPM and other boundary conditions, then called for SINDA/FLUINT to (1) advance its solution by the desired time value then (2) return compressor torques and other information, and (3) wait until another time step was needed.

GM's e-Thermal—In order to help designers size equipment such as air conditioning evaporators and transmission oil coolers, General Motors has developed a vehicle-level thermal/fluid simulation tool named e-Thermal (Ref 9). GM has created their own custom GUI which helps engineers that are not trained in SINDA/FLUINT build models comprised of both SINDA/FLUINT and in-house solutions. E-Thermal relies heavily on SINDA/FLUINT submodels and parametric modeling methods. The driveline model is based in MATLAB/Simulink.

EXCEL-BASED MODELING ENVIRONMENTS

Microsoft Excel™ and thermal software such as SINDA/FLUINT have enjoyed a long history together. For many years, especially before the advent of modern GUIs, creative users have employed Excel and other spreadsheet software to generate parts of thermal models (usually restricted to the network portion).

Also, in addition to various other plotting options, C&R provides a VBA and DLL-based Excel template that can be used to read and plot binary "save file" data (SINDA/FLUINT specialized format) in Excel, perhaps in order to export it to other more specialized plotting applications.

Over the years, several users and organizations have attempted to embed thermal solutions into Excel for the purposes of generating a preliminary design tool. One of the more recent such codes (and perhaps the only currently supported one) is Alstom's ThermXL (Ref 10).

Such tools require a translation step to export models to a fully-featured thermal analyzer capable of solving large models, but are usually unable to accept (import) external models. While such tools answer some needs, they leave others unresolved, including:

1. the ability to perform fluid system analysis, which is often more detailed than can be contained within VBA macros
2. connections to geometry including CAD data and FEM models

3. the ability to access and modify a fully complex thermal/fluid model including model-specific Fortran-style user logic and C-style expressions (again, beyond what can be encoded in VBA macros), and advanced solution drivers (such as optimization, reliability, parametric sweeps and design space scanning)
4. the ability to use Excel as a front-end only: as a custom GUI perhaps hiding the details of large and complex thermal/fluid models from non-experts
5. the ability to use Excel as a portal to other applications without requiring a limited or simplified thermal solution.

While the COM features in SINDA/FLUINT and Thermal Desktop have been available to advanced users for several years, they have not been formally documented since they are provided on a custom “as needed” basis. Recent work has changed that situation by the creation of Excel templates for launching, controlling, and communicating with both SINDA/FLUINT and Thermal Desktop (which may in turn invoke SINDA/FLUINT). Help files and examples are provided, and the templates are easily modifiable by advanced users (see Figure 3 for the two starting points).

Excel may now be used as a custom GUI for controlling arbitrarily complex thermal/fluid solutions. The user prepares a Thermal Desktop drawing or SINDA/FLUINT input file with suitable parameters (symbols and/or registers), and with appropriate calls to initiate the COM connection and data exchange. The Excel sheet can then be rearranged and expanded as needed.

For example, a loop heat pipe (LHP) model has been generated as shown in Figure 4 and fully described in Reference 11.

The LHP Excel model hides the underlying SINDA/FLUINT solution: a person not versed in SINDA/FLUINT can generate plots representing the steady-state or transient behavior of a custom (albeit simplified) LHP. The underlying model is itself available as a template for more detailed or customized SINDA/FLUINT models of LHPs, although other examples and templates are also available for that purpose as well.

Significantly, a licensed SINDA/FLUINT user can generate an Excel-based “prebuilt model” for use by others (including nonthermal specialists) without requiring further software purchases.* The unlicensed user of the Excel prebuilt can modify and rerun the analysis to whatever extent the SINDA/FLUINT user has provided for in his model or logic, as long as no changes to the SINDA/FLUINT input file are required.

Another example application is described in Reference 12: a preliminary design tool for spacecraft sizing and compo-

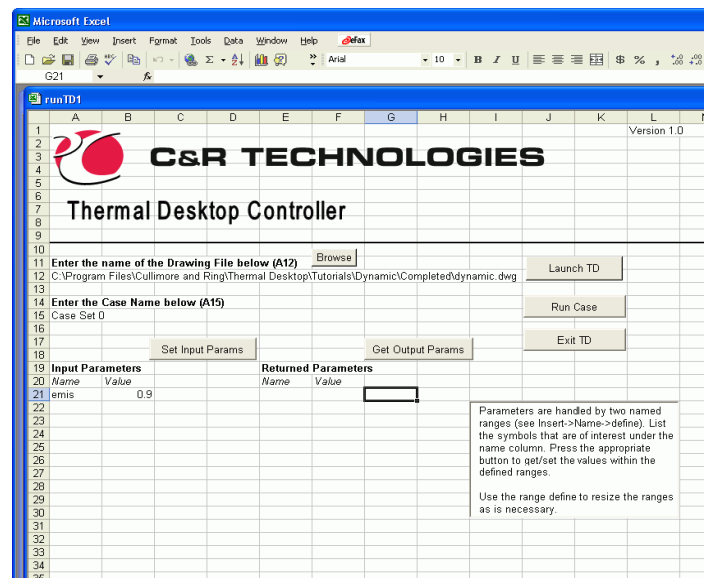
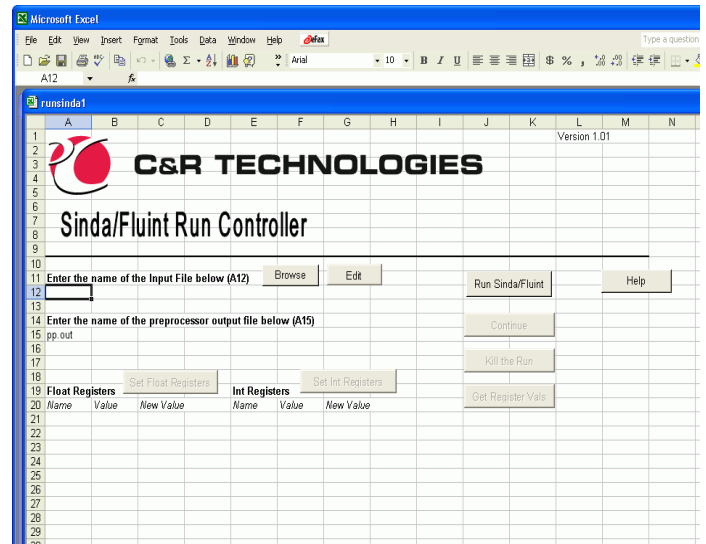


Figure 3: Templates for Controlling Either SINDA/FLUINT (top) or Thermal Desktop (bottom) from within Microsoft Excel™

nent location that can exploit the Excel interface to add data exchange to other third-party and in-house tools (e.g., structural, cost, etc.).

THE FUTURE

The existing COM interface allows exchange of character strings which might contain commands or single data values. Repeated calls, though inexpensive, are therefore necessary to complete a complicated (multi-step) operation or to exchange sufficient data. Combined with the ability to make extensive changes to a model via centralized parameters, many possibilities exist for “behind the scenes” integration with a variety of third party codes.

For example, a CFD calculation could utilize the piping network capabilities (or transient, two-phase, etc.) of SINDA/FLUINT as a side calculation. As another example, a pump

* To enable this capability, the free evaluation version of SINDA/FLUINT must be installed in order to set up Windows registers and DLLs, but no purchases are required.

This feature is no longer available with the free evaluation version

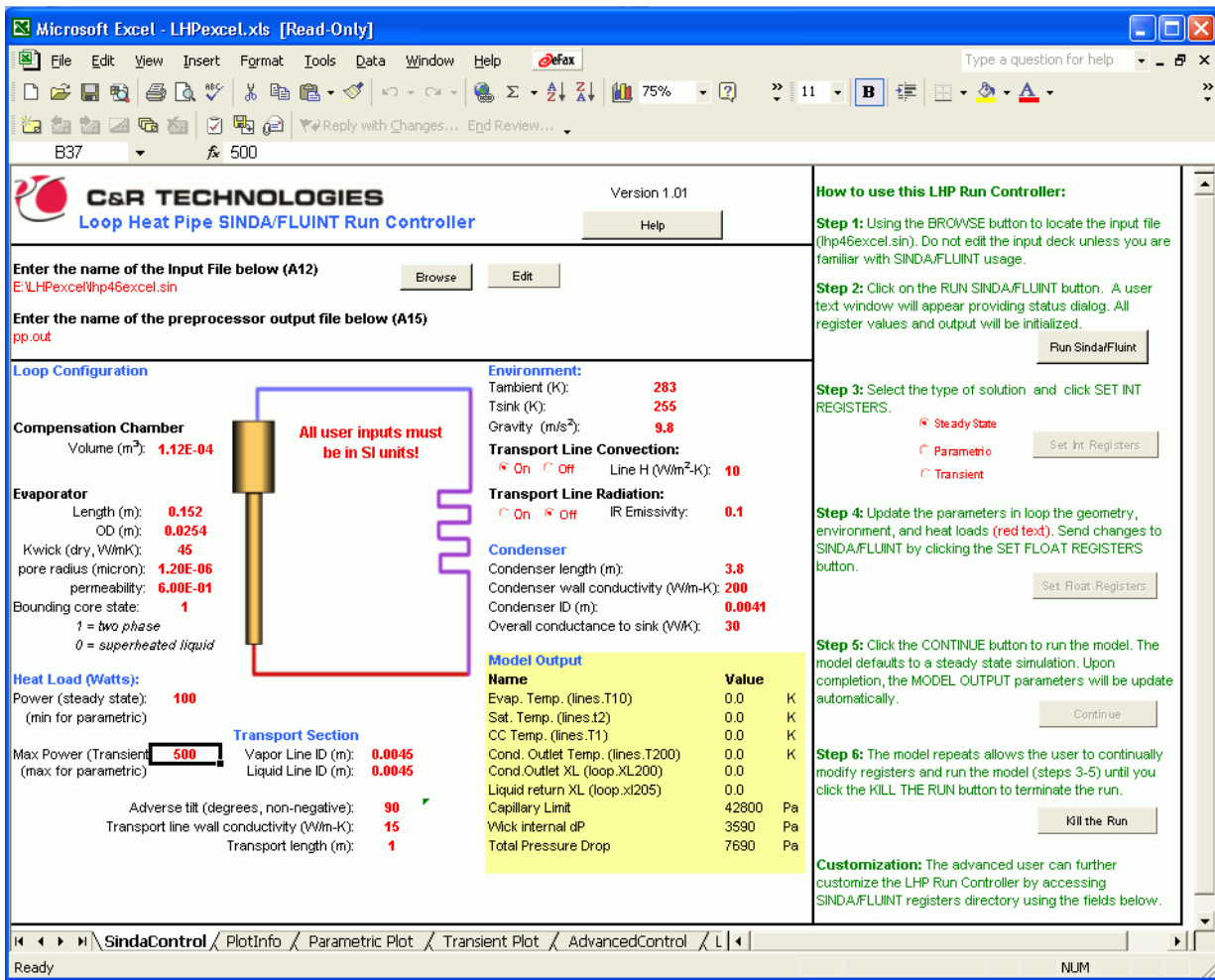


Figure 4: Excel-based LHP Model and “Prebuilt”

or compressor analysis code could be coupled to a “cycle code” like SINDA/FLUINT for purposes of sizing or for providing transient loading profiles.

However, there are limitations. First, extensive data, such as might be needed to couple geometric thermal data (e.g., flux distributions along a surface) to a CFD or aeroheating code, is too cumbersome to handle. Second, only a single connection can be established between processes. This means, for example, that only a single SINDA/FLUINT run can be launched and controlled at a time from a code such as Excel or MATLAB/Simulink. Third, COM connections do not exploit the existence of multiple processors, much less those distributed over networks.

C&R is therefore actively developing new technologies based on Microsoft’s .NET foundation. These new products will overcome all of the above limitations, allowing some or all of thermal/fluid solution technology to be embedded flexibly in a variety of applications. Some of this work (the next-generation SINDA) was funded under a recently completed SBIR contract (NAS8-02010) from the NASA Marshall Space Flight Center, and commercial deployment is imminent.

CONCLUSIONS

While parametric modeling options provide increased power and ease-of-use within stand-alone codes, even greater opportunities arise when that code is externally commandable.

Methods have been described for linking simple or detailed thermal/fluid models into third-party software. The purpose of creating such a link can be to provide a custom GUI, to exploit the unique capabilities of the third-party program (e.g., control systems in MATLAB, or CFD solutions), or to create simple links to multiple codes (e.g., using Excel as a common interface).

Consistent with the philosophy of SINDA, generalized mechanisms are provided without presuming specific applications, and therefore without limiting users’ creativity in making tools that are customized to their needs.

REFERENCES

- 1 T. Panczak, M. Welch et al, Thermal Desktop® User’s Manual, www.crtech.com.

- 2 B. Cullimore, "Optimization, Data Correlation, and Parametric Analysis Features in SINDA/FLUINT Version 4.0," SAE 981574, International Conference on Environmental Systems, July 1998.
- 3 B. Cullimore, "Nonlinear Programming Applied to Thermal and Fluid Design Optimization," ECTC/ITHERM proceedings, May 2002.
- 4 J. Baumann et al, "Nonlinear Programming Applied to Calibrating Thermal and Fluid Models to Test Data," SEMI-THERM proceedings, March 2002.
- 5 B. Cullimore, "Dealing with Uncertainties and Variations in Thermal Design", IPACK2001-15516, InterPack '01 Proceedings, July 2001.
- 6 B. Cullimore et al, "Integrated Analysis of Thermal/Structural/Optical Systems," SAE 2002-01-2444, July 2002.
- 7 B. Cullimore et al, "Automated Multidisciplinary Optimization of a Space-based Telescope," SAE 2002-01-2445, July 2002.
- 8 Hendricks, T. J., "Optimization of Vehicle Air Conditioning Systems Using Transient Air Conditioning Performance Analysis," Proceedings of the Vehicle Thermal Management Systems 5 Conference, Paper # 01VTMS-037, Nashville, TN, May 2001. See also www.ctts.nrel.gov/analysis/advisor.html
- 9 T. Tumas et al, "e-Thermal: A Vehicle-level HVAC/PTC Simulation Tool," SAE 2004-01-1510, March 2004.
- 10 www.techcentreuk.power.alstom.com/?pid=2
- 11 J. Baumann et al, "An Excel User Interface for Modeling LHPs in SINDA/FLUINT," SAE 2004-01-2555, July 2004.
- 12 D. Martin, "Parametric Models and Optimization Routines for Rapid Thermal Design," SAE 2004-01-2273, July 2004.

TD Thermal Desktop
 Thermal
 Desktop® CAD-based FDM/FEM thermal modeling environment from C&R Technologies
 TRASYS Thermal Radiation System
 VBA Microsoft Visual Basic for Applications
 .NET Microsoft communication framework

CONTACT INFORMATION

C&R Technologies, Inc.
 URL: www.crttech.com
 Voice: 303 971 0292
 Fax: 303 971 0035

DEFINITIONS, ACRONYMS, ABBREVIATIONS

CAD Computer Aided Design
 CFD Computational Fluid Dynamics
 COM Microsoft communication protocol
 COMINIT SINDA/FLUINT COM-initiation routine
 DLL Dynamic Link Library
 DOWAIT SINDA/FLUINT COM-wait routine
 ESA European Space Agency
 Excel™ Microsoft spreadsheet program
 FDM Finite Difference Modeling
 FEM Finite Element Modeling
 FloCAD® Fluid system analyzer in Thermal Desktop
 GM General Motors
 GUI Graphical User Interface
 iSIGHT® Multidisciplinary integration and optimization framework from Engineous
 LHP Loop Heat Pipe
 MATLAB® MathWorks' analysis system
 NREL National Renewable Energy Laboratory
 RadCAD® Radiation analyzer in Thermal Desktop
 SINDA/FLUINT Thermal/fluid analyzer from C&R Technologies
 SINDA Thermal side of SINDA/FLUINT
 Simulink® MATLAB simulation module