THE MODELLING OF THE THERMAL SUBSYSTEM IN SPACECRAFT REAL–TIME SIMULATORS

Miguel Á. Carreira Perpiñán

Simulation Systems Section

esa-esoc

3rd Workshop on Simulators for European Space Programmes

ESTEC, Noordwijk, The Netherlands
15-17 November 1994

Outline

- Outline and introduction
- Methods for thermal modelling
  - Integrators
    - Example: ESATAN
  - Interpolators
    - Example: real–time satellite simulators (RTSS) developed at ESOC/FCSD/SIM
- Suitability of integrators for RTSS
- Conclusions and future work

Introduction

- Primary functions of the RTSSs developed at ESOC/FCSD/SIM:
  - Testing and validation of the Operations Control Centre facilities and flight control procedures
  - Training of the ground–network operators
- The mission of the thermal subsystem of an RTS is to provide thermal telemetry (TTM) in due time
- Minimal requirements for the thermal subsystem:
  - Temperature of the equipment stays between certain margins
  - Temperature tendencies with time are more or less correct
- Approaches for the modelisation of a thermal system:
  - Integrators
  - Interpolators
  - Finite–element method: not suitable for real–time
- Heat transfer modes in a satellite:
  - Conduction
  - Radiation
  - Convection (less important)
- The payload is considered to be uncoupled with the rest of the system

Integrators

Steps:

1. **Nodalisation** of the thermal system, which produces a lumped parameter network (LPN) consisting of:
   - *Nodes*: isothermal volumes able to store heat [C_i (J/°C), Q_i (W), T_i (°C)]
   - *Links*: heat transfer paths [K_{ij} (W/°C), R_{ij} (W/°C^4)]
2. **Application of the heat transfer equations** (Q is the heat rate in watts):
   - Conduction: Fourier's law:

     $$Q = -\kappa A \nabla T$$

   - Radiation: Stefan–Boltzmann law for a grey body:

     $$Q = \sigma \alpha \epsilon \mathcal{F} A T^4$$

   and the principle of the conservation of energy:

   $$Q_{\text{in}} - Q_{\text{out}} = \Delta U = \Delta(\rho c V T)$$

   leads to the system

   $$C_i \frac{dT_i}{dt} = \sum_{i \neq j} K_{ij}(T_j - T_i) + \sum_{i \neq j} R_{ij}(T_j^4 - T_i^4) + Q_i, \quad i, j = 1, \ldots, n \tag{1}$$

Integrators (cont.)

3. **Integration** of system (1) w.r.t. the time:
   - Steady state: $\frac{dT_i}{dt} = 0$ for all i. Solve a system of nonlinear equations
   - Transient: finite–differences methods commonly use a first–order approximation for the time derivative

     $$\left(\frac{dT_i}{dt}\right)_t = \frac{T_i(t+\Delta t) - T_i(t)}{\Delta t} + \mathcal{O}(\Delta t^2)$$

     and compute $T_i(t_0 + (\Delta t)_1)$, $T_i(t_0 + (\Delta t)_1 + (\Delta t)_2)$, etc. known $T_i(t_0)$.

     Δt is the **timestep**.

Example of actual implementation:

- ESATAN (European Space Agency Thermal ANalyser), with ESARAD for the radiative calculations

Integrators (cont.)

Pros/cons:

- $\oplus$ High accuracy (for “small” Δt)
- $\oplus$ Also valid for non–nominal situations
- $\ominus$ High CPU consumption
- $\ominus$ Development of thermal model requires expertise

Interpolators

Steps:

1. Determination of parameters P_j that affect the S/C thermal behaviour. S/C thermal context: $\mathbf{C}(t) = (P_1(t), \ldots, P_p(t))$, $P_j(t) \in \mathbf{R}$
2. Context discretisation: $\tilde{\mathbf{C}} = (\tilde{P}_1, \ldots, \tilde{P}_p)$, $\tilde{P}_j \in \{\tilde{P}_{j,min}, \ldots, \tilde{P}_{j,max}\}$
3. Off–line computation with a thermal analyser of steady states for each $\tilde{\mathbf{C}}$:

$$\tilde{\mathbf{T}}_\infty : \{\tilde{P}_{1,min}, \ldots, \tilde{P}_{1,max}\} \times \cdots \times \{\tilde{P}_{p,min}, \ldots, \tilde{P}_{p,max}\} \longrightarrow \mathbf{R}^m$$
$$\tilde{\mathbf{C}} = (\tilde{P}_1, \ldots, \tilde{P}_p) \longmapsto \tilde{\mathbf{T}}_\infty(\tilde{\mathbf{C}}) = (\tilde{T}_{1,\infty}, \ldots, \tilde{T}_{m,\infty})^T$$

4. Interpolation for nondiscrete contexts: $\mathbf{T}_\infty(\mathbf{C}) = \mathcal{F}(P_1, \ldots, P_p)$
   - Global interpolation
   - Local interpolation among neighbouring discrete contexts
5. Real–time calculation of the S/C thermal map:

$$T_i(t + \Delta t) = T_{i,\infty} + (T_i(t) - T_{i,\infty})\, e^{-\frac{\Delta t}{\tau_i}}, \quad i = 1, \ldots, m$$

   The time constants τ_i are also determined with the help of a thermal analyser.

Interpolators (cont.)

Global interpolation

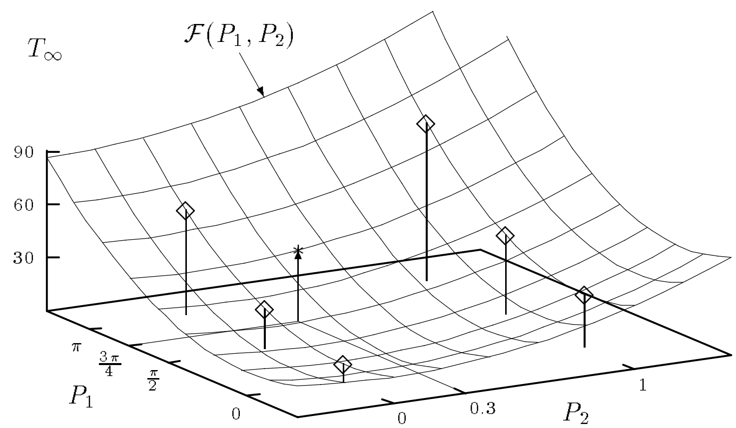

Figure 1: *Global interpolation for context* $\mathbf{C} = (\frac{3\pi}{4}, 0.3)$ *over parameters* $\tilde{P}_1 \in \{0, \frac{\pi}{2}, \pi\}$ *and* $\tilde{P}_2 \in \{0, 1\}$.

Local interpolation

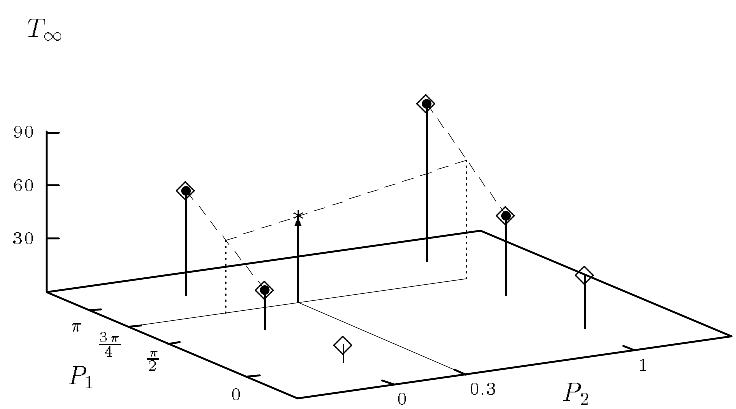

Figure 2: *Local bilinear interpolation for context* $\mathbf{C} = (\frac{3\pi}{4}, 0.3)$ *over parameters* $\tilde{P}_1 \in \{0, \frac{\pi}{2}, \pi\}$ *and* $\tilde{P}_2 \in \{0, 1\}$*, using 4 neighbours (highlighted).*

Interpolators (cont.)

Example of actual implementation:

- The thermal subsystem of the RTSs developed at ESOC/FCSD/SIM for the ISO, Italsat, Eureca and Pastel satellites (among others)

Pros/cons:

$\oplus$ Low CPU consumption

$\oplus$ Easy computation of steady states

$\oplus$ Straightforward implementation

$\ominus$ Low accuracy (interpolation error + inexact eq. for T_i)

$\ominus$ Extrapolation (for contexts out of those foreseen) produces uncertain predictions

$\ominus$ Necessary to store the tables that define $\tilde{\mathbf{T}}_\infty$

Use of integrators
in real–time satellite simulators

- Main reasons: better accuracy and applicability to non–nominal scenarios
- Involves two independent operations:
  - Actual integration of system (1), that provides with the S/C thermal map in due time
  - Update of the coefficients of system (1)

Performance of ESATAN solution routines

- Approximate formula for the CPU time in seconds employed by `SOLVIT` (steady–state by Newton–Raphson method):

$$\Delta t_{CPU} \approx \frac{t_i(n,c)i}{\text{MIPS}}$$

  where n is the number of nodes and c the number of links of the model, i the number of iterations required to solve the algebraic system (1), $t_i(n,c)$ the time per iteration and the CPU power is expressed in MIPS

- Objective: Minimise real–time fraction $\frac{\Delta t_{CPU}}{\Delta t}$
  - Increment Δt. Drawbacks: the approximation error $\mathcal{O}(\Delta t^2)$ grows; some routines may not converge (`SLFRWD`); TTM period is 1-10 seconds
  - Decrease $t_i(n,c)$ $\Leftrightarrow$ Decrease n and c in the LPN. Drawback: the model is less accurate.

    There exist algorithms for model reduction made by some companies, based on node grouping and node/link removal
  - Increase MIPS: currently, the RTSs at SIM run on VAX 4000.90 (30 MIPS); near future: DEC/Alphas ($>$ 100 MIPS)
  - i (`LOOPCT`): depends on the matrix of system (1) and on the desired accuracy of the iterative method. The SOHO and ISO models require about 100 iterations with `RELXCA` between 0.01 and 0.001

Performance of ESATAN solution routines (cont.)

- Assuming $i = 100$ and that 10% of the CPU time is allocated to the thermal solution routine:

n	c	Δt_{min}(30 MIPS)	Δt_{min}(100 MIPS)
50	300	9.7	2.9
100	1000	31	9.4
200	3000	93	28
500	10000	320	96

Table 1: *Minimum timestep in seconds for real–time simulation of ESATAN models.*

Update of the LPN coefficients

- Factors that influence the LPN coefficients:

  1. TCU command/telecommand: low frequency, not responsibility of the thermal subsystem
  2. Temperature: slow variations (great inertia)
  3. S/C geometry: great effect on R_{ij}
  4. S/C attitude and orbital coordinates: great effect on Q_i

  3. and 4. are usually computed by a ray–tracing algorithm, which can't run in real–time at all (minutes of CPU consumption) $\Rightarrow$ Table interpolation for R_{ij}, Q_i

Conclusions

- SIM's computer facilities allow for a hybrid method in real–time thermal simulation:
  - Numerical integration of a reduced ESATAN thermal model
  - Table interpolation for radiative couplings and external fluxes
- Benefits:
  - Improved accuracy of the generated thermal telemetry
  - Use of thermal mathematical models designed by thermal engineers right from the beginning
  - Reliable routines (ESATAN is validated through years of use)
  - Interpolation schemes can be simulated by an *ad–hoc* LPN with a clever election of coefficients

Future work

- ESATAN version tailored for real–time (?)
- Interface ESATAN $\leftrightarrow$ real–time simulator
- Powerful model reduction algorithms with error estimate