CHURCH-TURING

BACKGROUND

In order to set the scene for a response to Jon Smith's remarks it's worth recalling some of the underlying concepts underpinning Church- Turing. The Church-Turing theorem (or thesis since it has never been proved) is concerned with defining what is meant by an 'effective' or 'mechanical' method for, essentially solving, problems in logic and mathematics. In the Stanford Encyclopaedia of Philosophy (http://platto.stanford.edu/entries/church-turing/) the author of the article on Church-Turing (B. J. Copland) defines an 'effective' or 'mechanical' method or procedure 'M' as:

1. M is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols),
2. M will, if carried out without error, produce the desired result in a finite number of steps,
3. M can (in practice or in principle) be carried out by a human being unaided by any machinery save pencil and paper,
4. M demands no insight or ingenuity on the part of a human being carrying it out.

Turing, in looking for a 'computing engine' that can be organized in a way that encompasses these statements devised the Turing machine. He then put forward the proposition that a Turing machine can do anything that could be described as 'rule of thumb' or 'purely mechanical'. This is a way of explaining what is effectively calculable and Church, with whom Turing worked for a while at Princeton in the 1930's, devised another way of defining the concept. His thesis states that a function of positive integers is effectively calculable only if recursive. This, seems to me, to be saying that a function is effectively calculable if there exist (in practice or in principle) a computing engine controlled by an algorithm that can generate numerical values for the function.

CHURCH-TURING, FINITE ELEMENTS AND THE REAL WORLD

Moving on to the Finite Element Method and it's equivalence to the real world, consider the beam example displayed in page 222 of 'A Practical Guide to Reliable Finite Element Modelling'. Suppose we want to evaluate the vertical displacement of the beam shown in figure 8.8 on page 223. We could calculate it by way of a finite element analysis with an appropriate algorithm or set of algorithms and because the function is effectively calculable a set of numbers will be generated that describe the required displacement (shown in figure 8.8 which was generated using MathCad as the algorithm).

Alternatively these same numbers could be generated by performing an experiment on a real beam and taking measurements. In this situation the real world is performing the role of the computing engine (i.e. the function evaluator) and the algorithm with which it generates the required numbers is simply the process of measurement.

In the book the term K_R, the stiffness matrix of the real world, and K_N, the stiffness matrix of the finite element model, are two representations of the structure. These representations are appropriate to the two calculation engines that generate the numbers defining the beam displacement field. Clearly the terms in K_N are explicit since the matrix is created by the analyst and are directly available for observation whereas the terms in K_R are implicit but could be inferred from experimental observation is required.

In the experimental situation one might argue that statement 3 in the description of 'M' has been violated because measurement is not a process unaided by any machinery. However, as implied above, I would argue that measuring the displacement is part of the algorithm used by the calculating engine – after all the measurements could be taken purely mechanically without the need for human intervention.

This interpretation of the Church-Turing theorem (or thesis) is not unique to my book. Deutsch[1] claims that every finitely realizable physical system can be perfectly simulated by finite means. Wolfram[2] also states that the Church-Turing theorem says that any real world computation can be translated into an equivalent computation involving a Turing machine.

FINAL REMARKS

Jon has indicated that he considers the mathematical model K_M of K_R as the matrix that should be compared with finite element representation K_N for the purposes of applying the Church-Turing theorem. This implies that the proposition $K_M = K_N + e_N$ should replace the proposition $K_R = K_N + e_N$ used in the book. Clearly the arguments advanced in the section above show that I do not accept Jon's proposition. In fact K_M could replace K_N since the verification process should ensure that K_N converges to K_M. The problem of ensuring that K_N does indeed converge on K_M is addressed in the book. However, the really difficult problem is ensuring that K_M is making a realistic attempt to model K_R, which is the main issue addressed in the book

Alan Morris

Cranfield Friday, October 24, 2008

---

[1] Deutsch D 1985 'Quantum theory, the Church-Turing Principle and the Universal Quantum Computer' Proc Royal Soc, Series A, 400, 97-117.

[2] Wolfram S 2002 'A new kind of science' Champaign IL; Wolfram Media.