ALGEBRAIC COMPUTING IN GENERAL RELATIVITY

RAYMON A. d'INVERNO

Department of Mathematics, University of Southampton,
Southampton S09 5NH, England

Received 10 December 1974

ABSTRACT

The purpose of this paper is to bring to the attention of potential users the existence of algebraic computing systems, and to illustrate their use by reviewing a number of problems for which such a system has been successfully used in General Relativity. In addition, some remarks are included which may be of help in the future design of these systems.

§(1): INTRODUCTION

It is probable that the bulk of the time most theoretical physicists spend in research is taken up in performing routine algebra of one sort or another. By this is meant any form of algebraic manipulation of an algorithmic or semi-algorithmic nature such as addition, multiplication, substitution, differentiation, integration and the like. Perhaps there is no field in which this is more obvious than that of General Relativity. For example, a task very frequently encountered is that of computing curvature tensors and related quantities from a given metric. It was primarily for this reason that the author developed a computer system, originally called ALAM, which was specifically designed for carrying out many of the calculations encountered in General Relativity. Although algebraic systems have been used to date in tackling problems in Celestial Mechanics, Quantum Electrodynamics and Quantum Mechanics, it is in General Relativity where most use of them has been made.

The purpose of this paper is to bring to the attention of potential users the existence of algebraic computing systems and to illustrate their use by reviewing a number of the problems for which such a system has been successfully used by the author and colleagues. In addition some remarks are included which may be of help in the future design of algebraic computing systems. To this end it has been necessary to employ some technical computing language on occasions (especially in section 5), but it is hoped that the inclusion will not prevent the non-specialist from understand-
In this section we present a brief description of the system ALAM and a number of later versions of it. ALAM (Atlas LISP Algebraic Manipulator) was designed and constructed by the author within the Relativity research group at King's College, London. The terms of reference were such that a working system was required in a short a time as possible so that it could then be employed in research. It was partly for this reason that it was decided to base the system on a subset of a (high-level) programming language called LISP. Two of the additional reasons for this choice are first of all the fact that algebraic expressions can be naturally represented as so-called tree-structures or lists, and these are the basic data structures in LISP, and secondly LISP possesses a device called a garbage collector. This device is the housekeeper for the computer's store or memory and automatically throws away numbers or algebraic expressions when they are no longer required, enabling a program to continue which would otherwise run out of store and halt. This latter facility has proved to be something of a key requirement in large algebraic calculations.

There are two very important issues that arise in the construction of an algebraic computing system, and which to some extent were overlooked in the past when computers were primarily used for numerical work. The issues are efficiency and scope. By efficiency is meant the manner in which different systems perform with respect to the same problem, and by scope is meant the limit of the complexity of the calculational systems it can handle, in terms of what is economically viable. Even today there are many algebraic calculations which are beyond the limits of that which a machine can undertake. For these reasons, use was made of the largest, fastest computer then available, which was the Atlas 1. ALAM was then written by employing a version of LISP implemented on the Atlas, and hence its name. (In fact ALAM was written essentially in machine code and only employed the data structures, input/output routines and use of the basic system functions of LISP involving the stack and garbage collector).

ALAM possesses a number of distinctive features (for details see [1]), chief among which is the strategy of simplification. Simplification is the most important activity of an algebraic computing system, and in ALAM, it mirrors to a large extent what one actually does in a hand calculation. Briefly, the simplification procedure consists of three steps: an algebraic expression is truncated by eradicating all zeros occurring in it, then the expression is expanded using the laws of distributivity and associativity, and finally each term in the resulting expression is simplified according to the usual rules and like terms are collected together. This rather ad hoc, but straightforward, procedure turned out to work very efficiently. Another feature of the system is that it prints out results in a three-line mathematical format (the first system