

# Càlcul simbòlic per ordinador

Pere Castellví\* i Xavier Jaén†

## Introducció

Durant molts anys, per a la majoria de físics, l'expressió "càlcul per ordinador" ha estat sinònim de "càlcul numèric per ordinador". En efecte, amb l'aparició dels primers ordinadors cap a meitat de la dècada dels quaranta, el càlcul numèric, que havia estat inventat molts anys abans, va experimentar un important impuls. Els científics i enginyers van trobar en els ordinadors l'instrument adequat per dur a terme càlculs aproximats i van començar a incorporar aquesta disciplina, el càlcul numèric per ordinador, a les seves activitats. Aquest procés es va veure afavorit per la creació, cap a meitat dels anys cinquanta, del llenguatge de programació FORTRAN (FORMULA TRANslation) que, com el seu nom indica, va ésser especialment dissenyat per IBM per treballar amb fórmules matemàtiques. Des d'aleshores, i durant gairebé mig segle, el càlcul numèric ha progressat com a branca de les matemàtiques, ha estat present en qualsevol àrea de la ciència i la tecnologia i s'ha popularitzat com a eina didàctica.

Tanmateix, és ben conegut que una part important dels càlculs que es realitzen a la física no són aproximacions numèriques, sinó càlculs exactes de l'estil de derivar una expressió, substituir unes variables per unes altres i simplificar i reordenar els resultats. Aquesta mena de càlculs s'anomenen *algebraics* o *simbòlics* perquè, generalment, consisteixen a manipular una sèrie de símbols que poden representar nombres, però també polinomis, matrius, funcions, tensors o el que sigui. Exemples elementals de transformacions que apareixen sovint en efectuar càlculs simbòlics són

$$(1 + x)^2 \rightarrow 1 + 2x + x^2,$$

$$\frac{d}{dx} \sin^2 x \rightarrow 2 \sin x \cos x.$$

En alguns camps de la física, aquestes manipulacions poden arribar a ser extremadament feixugues i la probabilitat de cometre algun error s'incrementa notablement.

\*Pere Castellví Gironès (Reus, 1960) és llicenciat en Física per la Universitat de Barcelona (1985). Actualment és professor de secundària i del Dept. de Física Aplicada de la UPC.

†Xavier Jaén Herbera (Barcelona, 1958) és doctor en Física per la Universitat de Barcelona (1988). Actualment és professor del Dept. de Física i Enginyeria Nuclear de la UPC. Actualment els autors estan desenvolupant un paquet de càlcul tensorial basat en el MATHEMATICA. (e-mail jaen@fen.upc.es)

La idea d'utilitzar els ordinadors per dur a terme aquest tipus de càlculs és tant o més antiga que els mateixos ordinadors, però no s'ha fet realitat fins fa poc, quan la potència dels microordinadors ha esdevingut notable. Així, en els darrers anys, s'ha començat a produir la difusió d'una nova categoria de *software* que permet realitzar càlculs simbòlics, del tipus abans esmentat, amb ordinador. Aquesta modalitat de càlcul matemàtic amb ordinador rep una àmplia varietat de noms en anglès: *algebraic computation*, *symbolic computation*, *formula manipulation* i *computer algebra*, per citar-ne alguns. En els últims temps, però, sembla que comença a imposar-se la denominació *computer algebra* i els programes reben el nom de *computer algebra systems*. La traducció més estesa en català, i que seguirem aquí, és *computació algebraica* (CA) i *sistemes de computació algebraica* (SCA).

Els SCA es poden classificar en *sistemes d'àmbit genèric* (d'ara en endavant, SAG) i *sistemes d'àmbit específic* (SAE) segons que estiguin pensats per efectuar una gamma àmplia de càlculs o per resoldre un problema específic. Actualment, la majoria dels SAE es programen basant-se en un SAG i aleshores es diu que són un *paquet* (de l'anglès *package*). Per exemple, MATHEMATICA és un dels moderns SCA d'àmbit genèric i hi ha nombrosos paquets que amplien les prestacions d'aquest sistema. Aquests paquets s'han programat utilitzant el llenguatge de programació de MATHEMATICA i abasten temes tan variats com ara l'estadística, els gràfics en tres dimensions o el càlcul vectorial.

Si bé l'ús generalitzat dels SCA per efectuar càlculs simbòlics s'ha començat a produir a principis dels noranta, els seus orígens arrenquen de molt més enrere i estan íntimament lligats a la física teòrica.

## Història dels SCA

No intentarem fer un repàs exhaustiu de la història de la computació algebraica sinó només esmentar els fets que han resultat decisius per al desenvolupament posterior d'aquesta disciplina. De la mateixa manera, únicament els sistemes i paquets que han tingut una implantació important seran objecte de comentari.

Cap al principi dels seixanta apareix LISP (LIST-Processing), un llenguatge de programació d'alt nivell concebut per manipular dades agrupades en llistes (de números, de caràcters o de cadenes de caràcters). Per la seva pròpia naturalesa, LISP facilita la implementació



d'operacions de càlcul simbòlic, com ara la derivació de funcions o el producte de polinomis. Si FORTRAN havia estat l'eina ideal per al càlcul numèric, LISP ho seria per al càlcul simbòlic i alguns dels SCA més importants s'escriurien en aquest llenguatge.

Una de les primeres aplicacions en LISP fou SAINT (Symbolic Automatic INtegration), ideat per efectuar integració simbòlica i amb un rendiment similar al d'un estudiant universitari de primer curs. Cap al 1964, IBM va desenvolupar FORMAC, una extensió de FORTRAN per al processament de llistes que afegia unes 15 instruccions per efectuar manipulació simbòlica de polinomis, funcions racionals i funcions elementals i que no va assolir l'èxit esperat per problemes de gestió de memòria. Aquests sistemes, i alguns més, com ara MATHLAB, PM (Polynomial Manipulation) i ALPAK integren el que es podria considerar la primera generació de SCA. En general, se'n va fer un ús limitat i la seva principal contribució ha estat la de constituir el banc de proves sobre el qual s'han elaborat altres sistemes més evolucionats.

L'any 1963, Anthony C. Hearn, un físic de partícules, seguint un suggeriment del creador de LISP, John McCarthy, comença a treballar en el disseny d'un programa per calcular a l'electrodinàmica quàntica utilitzant LISP com a llenguatge de programació. El 1966 apareix la primera publicació descrivint els resultats assolits en aquest terreny: càlculs que un científic tardaria sis mesos a fer-los a mà estan enllestits en menys de quinze minuts. A partir d'aquí, el programa, que havia nascut com un SAE, evoluciona cap a un SAG i el 1968 es publica un article que descriu el nou producte: REDUCE. Aquest nom no és un acrònim sinó que intenta reflectir un dels objectius que va perseguir el seu autor en el desenvolupament del programa: reduir la complexitat de les expressions que apareixen en utilitzar un SCA.

REDUCE va ser ideat com un *sistema interactiu*, en el qual l'usuari formula una pregunta al sistema mitjançant un *input*, el sistema respon amb un *output* i segons la resposta l'usuari actua en conseqüència, de la mateixa manera que ho faria amb una calculadora electrònica de butxaca. Actualment, tots els SCA treballen d'aquesta manera, però en el moment de desenvolupar REDUCE, el *modus operandi* més comú era ben diferent: hom agrupava en un programa tots els càlculs que es volien fer, s'executava aquest programa en l'ordinador del centre de càlcul —pel qual s'havia reservat hora prèviament— i, finalment, hom recollia tots els resultats. Habitualment, no quedava temps per analitzar els resultats i tornar a executar el programa si alguna cosa no havia funcionat, de manera que s'havia d'esperar fins a la propera reserva de l'ordinador.

El 1970 va aparèixer la versió 2 de REDUCE i el 1983 la versió 3, aquesta última incloent una sèrie de paquets per a la integració analítica, factorització de polinomis amb més d'una variable, aritmètica real amb precisió ar-

bitrària i resolució d'equacions que havien estat creats per usuaris del programa. De fet, REDUCE sempre ha estat un sistema obert, en el sentit que és possible obtenir el codi LISP complet en adquirir el programa. Aquest aspecte de REDUCE ha originat la col·laboració dels usuaris en el desenvolupament i millora del programa, ha facilitat la incorporació de nous algorismes tan bon punt han estat desenvolupats i n'ha permès la utilització com a eina educacional en el camp de la CA.

Una altra de les característiques importants de REDUCE és la seva *portabilitat*, és a dir, la possibilitat de ser utilitzat en diferents màquines amb mínimes modificacions en el codi del programa. En primera instància, això s'aconseguí gràcies al fet que LISP estava disponible en un elevat nombre d'ordinadors i més tard, quan LISP va evolucionar cap a diferents dialectes es va reescriure el programa utilitzant un subconjunt de LISP que fos segur de trobar-lo en tots els dialectes. Una conseqüència directa de l'alta portabilitat de REDUCE és l'àmplia difusió que ha assolit i la proliferació de paquets especialitzats en diverses àrees de la física i les matemàtiques generats pels mateixos usuaris.

L'última versió comercialitzada de REDUCE és la 3.5 i va aparèixer l'octubre de 1993. Entre les prestacions que ofereix trobem operacions amb polinomis i funcions racionals, simplificacions de forma automàtica o controlada per l'usuari, càlculs simbòlics amb matrius, aritmètica amb enters i reals de precisió arbitrària, possibilitats de definir noves funcions, derivació i integració analítiques, factorització de polinomis, resolució d'una àmplia varietat d'equacions algebraïques, facilitats per generar programes numèrics a partir d'*inputs* simbòlics i càlculs amb matrius de Dirac per a la física de partícules.

Si bé en els darrers anys la majoria de SCA han integrat en un mateix entorn càlculs simbòlics, numèrics i capacitats gràfiques, REDUCE ha optat per mantenir la idea original i restringir els serveis que ofereix a l'àmbit estricte de la CA, proporcionant facilitats per escriure els resultats en un format que pugui ser entès per altres programes de càlcul numèric o de representacions gràfiques. Els avantatges d'això són que permet concentrar-se en un únic aspecte i vendre el programa a preus moderats.

Com veurem, els moderns SCA han heretat algunes de les característiques que han fet de REDUCE un sistema força utilitzat: interacció, portabilitat, un llenguatge de programació i la possibilitat d'ampliació mitjançant paquets.

Al voltant de 1971 va aparèixer MACSYMA (Project MAC's Symbolic Manipulation system), un SAG programat en LISP i preparat per efectuar càlculs simbòlics i numèrics. Va ser concebut com un potent sistema capaç de derivar, integrar, calcular límits i sèries de Taylor, resoldre sistemes d'equacions polinòmiques i equacions diferencials i treballar amb matrius i tensors. De la ma-



teixa manera que REDUCE, MACSYMA era un sistema interactiu i incorporava un llenguatge de programació. Per poder desenvolupar tota la seva potència MACSYMA necessitava ordinadors amb importants recursos tant pel que fa a capacitat de disc com a memòria RAM i això el va fer de difícil accés i va aconseguir molta menys difusió que REDUCE. La recent aparició d'una versió per a PC, que els seus distribuïdors anuncien com el millor sistema actual, ha fet que es renovi l'interès per aquest sistema. Permet la resolució d'equacions diferencials ordinàries i en derivades parcials, tota classe de gràfics en 2D i 3D, animació, traducció de programes escrits en MACSYMA a C i FORTRAN i ajut a través d'hipertext.

REDUCE i MACSYMA (en les seves versions originals) són els representants més genuïns de la segona generació de SCA i si REDUCE va ser el més utilitzat, MACSYMA va ser, amb diferència, el més potent i el que més incidència va tenir pel que fa a desenvolupament d'algorismes. De fet, moltes de les tècniques estàndard que s'utilitzen actualment han estat desenvolupades o directament influïdes per les investigacions del grup de científics que van crear MACSYMA al MIT. Hi ha molts altres SCA que es poden considerar de la segona generació, entre ells alguns SAG com SCRATCHPAD i alguns SAE com CAMAL (originàriament dirigit a la mecànica celeste i posteriorment utilitzat a la relativitat general), SHEEP (càlcul tensorial), TRIGMAN (sèries de Poisson), SCHOONSCHIP (partícules elementals) i EXCALC (càlcul exterior). Alguns d'aquests sistemes han anat evolucionant amb el temps i, actualment, encara s'utilitzen.

## La tercera generació

Els SCA de la segona generació, i particularment MACSYMA, es caracteritzaven per necessitar molt espai d'emmagatzemament en disc i molta memòria RAM, de manera que durant els anys seixanta i setanta hi havia pocs ordinadors amb recursos suficients per poder executar els sistemes existents a l'època. Aquesta va ser la causa que l'ús d'aquests sistemes es limités a aquells investigadors que tenien accés a grans ordinadors. Amb l'aparició, cap al principi dels vuitanta, dels microprocessadors la situació va canviar. Ara era possible obtenir una estació de treball capaç d'executar un SCA i disposar-ne lliurement, sense les restriccions d'un centre de càlcul. L'impuls definitiu el va donar la irrupció en el mercat dels PC, que posava a l'abast de quasi tothom ordinadors prou potents a un preu assequible.

El primer SCA a córrer en PC va ser MUMATH, desenvolupat per D. Stoutmeyer i A. Rich a la Universitat de Hawaii al final dels setanta. Va ser escrit en LISP i incloïa un llenguatge de programació propi anomenat MUSIMP. MUMATH es podia executar des d'un disc flexible de 360 kbytes en ordinadors IBM-PC o compatibles amb 640 kbytes de memòria RAM i 4,7 MHz de velocitat (aquesta era la configuració més comuna a meitat

de la dècada passada). Encara que les seves possibilitats eren modestes, MUMATH es va fer força popular pel fet de poder-se utilitzar en qualsevol PC.

L'any 1971, Dennis Ritchie dels Bell Laboratories va dissenyar el llenguatge de programació C i juntament amb Ken Thompson el va utilitzar per escriure el sistema operatiu UNIX, trencant amb la tradició d'escriure els sistemes operatius amb llenguatge d'assemblador. Amb el temps, tant C com UNIX van adquirir celebritat i, al principi dels vuitanta, els SCA es van començar a escriure en C, aprofitant-ne la portabilitat.

MATHEMATICA és un SCA genèric desenvolupat per la Wolfram Research. El creador d'aquesta empresa, Stephen Wolfram, durant el període 1980-81 va desenvolupar SMP (Symbolic Manipulation Program), el SCA que havia d'ésser el predecessor de MATHEMATICA. Discrepàncies sobre els drets de propietat intel·lectual de SMP amb els administradors de Caltech el van portar primer a Princeton i després a Illinois, on fundà la Wolfram Research l'any 1987. Aprofitant les idees de SMP que millor havien funcionat, va acabar de desenvolupar MATHEMATICA i la versió 1 del programa es va anunciar oficialment el 23 de juny de 1988. El mes de juny de 1991 va aparèixer la versió 2.

MATHEMATICA fou el primer SCA que integrà càlcul simbòlic, càlcul numèric i capacitats gràfiques en un mateix programa de forma unificada, amb l'objectiu, segons el propi autor, de "construir una eina molt general que fos útil en qualsevol tipus de càlcul tècnic". Ha estat escrit en C i està constituït per dues parts: el *kernel* (nucli) i el *front end* (la façana). El *kernel* és sempre el mateix, independentment de la màquina en la qual s'executi; és la part encarregada de dur a terme els càlculs i el fet que hagi estat escrit en C n'assegura la portabilitat. El *front end* s'ocupa de l'intercanvi d'informació entre el *kernel* i l'usuari, és específic per a cada ordinador i pot tenir dos aspectes diferents segons que l'entorn de treball suporti un monitor de text o gràfic. Les versions basades en interfícies gràfiques generen uns documents anomenats *notebooks* que permeten combinar text, ordres del MATHEMATICA i gràfics (que poden animar-se) i que poden utilitzar-se amb caràcter pedagògic o per efectuar càlculs.

A part de realitzar les tasques més comunes de la computació algebraica com poden ser operacions algebraiques i simplificació amb funcions elementals, MATHEMATICA és capaç de trobar primitives, resoldre equacions i sistemes de forma exacta, obtenir sèries de potències, calcular límits, integrar equacions diferencials, efectuar l'anàlisi estadística d'una sèrie de mesures, diagonalitzar una matriu, interpolat una funció per un conjunt de punts, efectuar una integració numèrica i fer representacions gràfiques de funcions en dues i en tres dimensions, per citar algunes de les possibilitats. Incorpora les definicions de multitud de fun-



cions que apareixen freqüentment a les matemàtiques i la física: funcions de Bessel, funció error, funció gamma, funció hipergeomètrica, polinomis de Legendre, harmònics esfèrics, polinomis d'Hermitte, funcions el·líptiques, tot tipus de distribucions estadístiques i moltes altres funcions. A més de les possibilitats simbòliques, numèriques i gràfiques que hem esmentat abans, MATHEMATICA incorpora un llenguatge de programació d'alt nivell i estructurat que el converteix en la plataforma ideal per desenvolupar aplicacions més específiques. En aquest sentit, el sistema pot ser ampliat posant les definicions necessàries en un fitxer a part, un paquet, i carregant-lo des de MATHEMATICA.

Actualment, MATHEMATICA és el SCA de més penetració en el mercat, s'han escrit un gran nombre de llibres dedicats a explicar-ne el funcionament i algunes de les possibles aplicacions. Hi ha una revista, *The Mathematica Journal*, de periodicitat trimestral, dedicada només a notícies i articles relacionats amb el MATHEMATICA, una base de dades electrònica, *Mathsource*, amb *notebooks* i paquets a la qual es pot accedir gratuïtament i s'han creat diferents grups d'usuaris per intercanviar informació sobre el programa.

Al principi de la dècada dels vuitanta apareix MAPLE, iniciat per Gaston Gonnet i Keith Geddes de la Universitat de Waterloo (Ontario, Canadà) i desenvolupat pel Symbolic Computation Group de la Universitat de Waterloo amb l'objectiu de fer accessibles els SCA a un major nombre d'usuaris dels que havien tingut durant la dècada passada. Actualment, MAPLE funciona en ordinadors que van des dels *mainframes* fins als ordinadors personals tipus IBM-PC o Macintosh passant per les estacions de treball. MAPLE no és un acrònim, sinó que fa referència a l'origen canadenc del programa (el símbol que apareix a la bandera del Canadà és una fulla d'auró, *maple* en anglès). De manera similar al MATHEMATICA, MAPLE ha estat dissenyat amb estructura modular i consta de tres parts: el *kernel*, el nucli de programa, l'*Iris*, la interfície amb l'usuari i la biblioteca de funcions i aplicacions. L'*Iris* i el *kernel* constitueixen una petita part del total del sistema, han estat escrits en C i ambdós es carreguen automàticament quan s'inicia una sessió amb MAPLE. El *kernel* interpreta les entrades de l'usuari, s'encarrega de les operacions matemàtiques més elementals i d'aquelles que el seu temps d'execució és crític i realitza la gestió de memòria. L'*Iris* s'ocupa de les comunicacions entre l'usuari i el *kernel*, de la detecció i notificació d'errors en les entrades, de la presentació de les sortides i del traçat de gràfics.

MATHEMATICA i MAPLE són els dos SCA d'àmbit genèric més utilitzats en aquests moments, per bé que ara mateix MACSYMA ofereix unes prestacions similars i que la ràpida evolució que està tenint el *hardware* l'ha fet més assequible. Podem dir que aquests dos sistemes encapçalen la tercera generació de SCA. El tercer

sistema pertanyent a aquesta generació que val la pena esmentar és DERIVE, l'hereu directe de MUMATH. La característica principal de DERIVE és el seu sistema de menús des del qual es pot accedir a totes les funcions. Disposa d'instruccions per efectuar càlcul diferencial i vectorial, resoldre equacions i sistemes, treballar amb matrius, representar gràfics en dues i tres dimensions i té un petit llenguatge de programació. Actualment es comercialitza la versió 3.0, la qual només necessita un PC compatible amb processador 8086 i un disquet de doble densitat per funcionar. És un sistema que pot resultar ideal per a estudiants, tant pel seu ús, intuïtiu i simple, com pel seu preu.

A més dels sistemes que hem comentat aquí, que són els més destacats, hi ha un nombre elevat de SCA tant d'àmbit genèric com especialitzat, alguns dels quals es poden obtenir gratuïtament. El lector interessat podrà trobar a la referència (Christensen, 1994) una llista de bona part d'aquests sistemes amb les indicacions necessàries per aconseguir-los.

## La CA com a ciència

Paral·lelament al progrés dels SCA s'ha produït el desenvolupament de la computació algebraica com a ciència. En efecte, a mesura que s'anaven desenvolupant nous SCA es feia patent la necessitat d'algorismes que resolguessin determinats problemes. Alguns d'aquests algorismes ja existien i només s'havien d'adaptar per tal que resultessin més eficients o que consumissin menys memòria, mentre que d'altres va ser necessari inventar-los. Molts d'aquests algorismes són extraordinàriament laboriosos i resultarien del tot impracticables per a un ésser humà, de forma que mai no s'haurien arribat a utilitzar sense l'existència dels ordinadors i dels SCA. L'exemple més paradigmàtic és l'algorisme de Risch (1968), que descriu un procediment per decidir si és possible trobar una primitiva d'una funció elemental en termes de funcions elementals<sup>1</sup> i, en cas que la resposta sigui afirmativa, diu la forma de trobar-la (Geddes, 1992). Els moderns SCA com ara MATHEMATICA i MAPLE incorporen aquest algorisme, de manera que, si no obtenim resposta per a la primitiva d'una funció construïda a partir de funcions elementals, podem estar segurs que és perquè aquesta primitiva no existeix.

Com que es tracta d'una disciplina relativament jove, bona part dels articles d'investigació sobre CA es troben dispersos en revistes d'informàtica, matemàtiques i física, per bé que comencen a haver-hi revistes dedicades exclusivament a aquest tema, com pot ser *Journal of Symbolic Computation*, i es poden trobar alguns llibres que tracten els aspectes de la CA fermament

<sup>1</sup>Recordem que el conjunt de funcions que comunament s'anomenen elementals inclou les funcions racionals, potencials, exponencials, logarítmiques, trigonomètriques i combinacions de totes aquestes i, per exemple, no són funcions elementals les funcions de Bessel o la funció error.



establerts. També s'han organitzat conferències, primer per separat a Amèrica del Nord i a Europa i a partir de l'any 1988 conjuntament a tot el món organitzades per l'ISSAC (International Symposium on Symbolic and Algebraic Computation).

D'altra banda, el nombre d'àrees d'aplicació de la CA ha experimentat un increment notable. Si, inicialment, els SCA s'utilitzaven sobretot en relativitat, mecànica quàntica i mecànica celeste, amb el temps s'han començat a fer aplicacions en mecànica de fluids, física de plasmes, òptica, mecànica clàssica i mecànica dels medis continus, sense tenir en compte les aplicacions pròpies dels matemàtics i algunes de la química teòrica.

En els darrers anys també s'han començat a utilitzar els SCA com a eina educativa gràcies, en part, a les possibilitats gràfiques que incorporen els sistemes de la tercera generació. Hi ha cursos de càlcul basats exclusivament en SCA (Brown, 1991) i multitud de llibres de diverses matèries (estadística, equacions diferencials en derivades parcials, mecànica quàntica, etc.) que d'una manera o altra utilitzen la CA.

## L'ús dels SCA

Pel que fa a la utilització dels SCA, és possible distingir dos nivells diferents, segons el tipus de problemes que es vulguin tractar. El primer nivell —el més elemental— és el que en podríem dir el de l'usuari ocasional. En aquest sentit, unes mínimes nocions sobre la sintaxi del programa són suficients per poder utilitzar el SCA ràpidament d'una forma interactiva, com si fos una calculadora simbòlica i fer operacions simples com ara calcular una derivada o una integral, ajustar una funció a unes dades experimentals pel mètode dels mínims quadrats, obtenir els valors propis i vectors propis d'una matriu, representar el gràfic d'una funció, resoldre una equació, obtenir la sèrie de Taylor d'una funció o calcular la mitjana i la variància d'una sèrie estadística. Només per poder fer aquest tipus de manipulacions algebraïques, numèriques i gràfiques de forma ràpida i sense possibilitat d'error, ja val la pena d'utilitzar un SCA i qualsevol que s'acostuma a fer-ne servir un, difícilment sap prescindir-ne. De fet, tot indica que d'aquí a uns anys els SCA s'hauran convertit en un instrument tan habitual com ho és ara una calculadora científica.

El segon nivell d'aprenentatge implica un coneixement més profund del SCA i permet afrontar problemes que es caracteritzen per la seva major dificultat i pel fet que es poden resoldre de forma algorísmica, seguint sempre els mateixos passos. Dos exemples de problemes de dificultat moderada són: (1) donat un sistema mecànic representat per un lagrangià  $L(q_i, \dot{q}_i, t)$ , obtenir les equacions de moviment del sistema i (2) donat un tensor simètric de rang 2, obtenir una rotació dels eixos de coordenades que el diagonalitzi. Per resoldre aquest tipus de problemes de la forma més automatitzada pos-

sible és convenient programar una funció o un paquet, segons la complexitat. L'aprenentatge de les tècniques de programació necessàries per fer això és relativament dur i pot no interessar a l'usuari ocasional. Amb tot, moltes vegades es dona el cas que la funció que es necessita ja ha estat programada per un altre usuari i posada a disposició pública en alguna de les nombroses bases de dades electròniques d'universitats de tot el món a les quals es pot accedir fàcilment a través de la xarxa Internet.

Per adquirir un coneixement superficial del programa, però suficient per encarar problemes del primer nivell, és aconsellable començar amb textos introductoris (Blachman, 1992; Heck, 1993; o Rayna, 1987). En molts casos, en aquests llibres es poden trobar exemples, suggeriments i exercicis que ajuden força en un primer aprenentatge. En canvi, a l'hora de programar es convenient disposar d'una referència amb una informació exhaustiva i el més acurada possible i llavors el més indicat són els manuals.

Els SCA també presenten algunes limitacions i dificultats inherents a la seva pròpia naturalesa. Un dels temes clau i gens trivial és el de la simplificació. Quan es realitzen manipulacions algebraïques a mà, en cada pas es procedeix de la forma que creiem més convenient en aquell moment, seguint a vegades uns criteris i a vegades uns altres de diferents, sempre amb l'objectiu de deixar les expressions en la forma més simplificada possible. Per contra, els SCA s'ajusten sempre als mateixos procediments en la manipulació d'expressions i les transformen fins a deixar-les en *forma normal*.<sup>2</sup> Per exemple, els polinomis tenen dues formes normals: completament expandits i completament factoritzats i els SCA incorporen funcions que permeten transformar entre una forma i l'altra. En canvi, difícilment aconseguirem efectuar la transformació

$$x^3 + 3x^2 + 3x \rightarrow (x + 1)^3 - 1$$

amb un SCA ja que el polinomi del segon membre està en una forma mixta: ni completament factoritzat, ni completament expandit. Si volem dur a terme simplificacions d'aquest estil hem d'ajudar nosaltres al sistema. De la mateixa manera, sorprèn enormement que un SCA no sigui capaç de realitzar una simplificació tan òbvia com ara

$$\sqrt{x^2} \rightarrow x,$$

i és que quan nosaltres fem aquesta simplificació estem utilitzant una informació addicional de la qual el programa no disposa i sense la qual el procediment no seria correcte. Dificultats d'aquest estil es presenten freqüentment i el més indicat en aquests casos és donar un cop de mà al programa per tal d'arribar a l'expressió buscada.

<sup>2</sup>Per a una explicació més extensa d'aquest terme podeu consultar Geddes (1992)

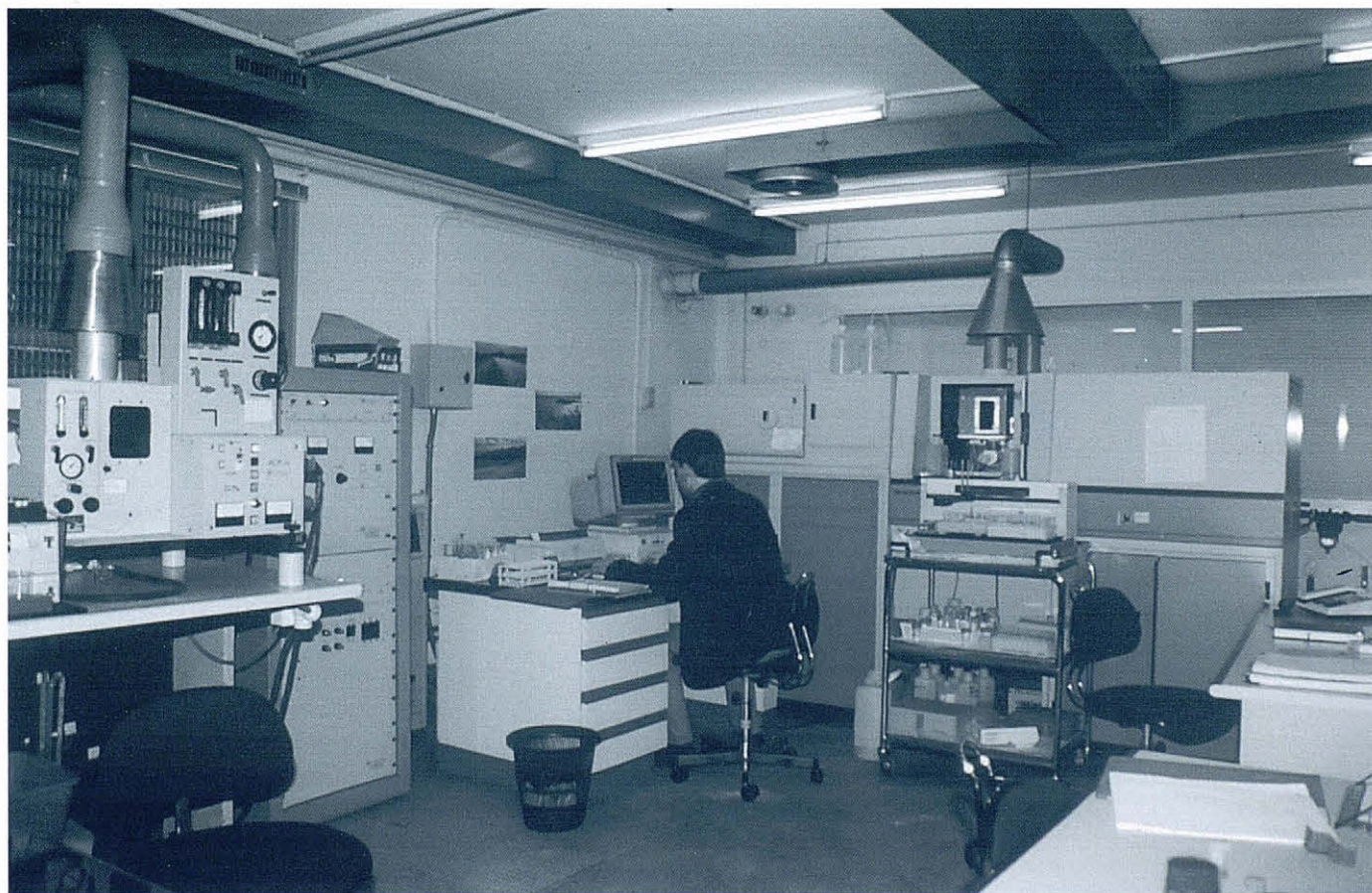


Un segon aspecte, relacionat amb la simplificació d'expressions, és la dificultat per identificar en termes d'objectes matemàtics la informació apareguda pel monitor. Si la complexitat del problema objecte d'estudi és moderada o gran, pot passar perfectament que la resposta del sistema a la nostra entrada —després de simplificada— ocupi unes quantes pantalles de l'ordinador. Això, a part de l'efecte desconcertant inicial, suposa in-

vertir un cert temps i esforç a desxifrar el contingut de la sortida. Generalment, aquest inconvenient es veu agreujat pel fet que el sistema només pot utilitzar un conjunt molt reduït de símbols que no inclou l'alfabet grec ni la majoria de símbols matemàtics i, per tant, les magnituds que estem acostumats a veure escrites d'una manera se'ns presenten de forma diferent, afegint dificultat a la lectura de la resposta.

## Referències

- BLACHMAN, N., *Mathematica, a Practical Approach.*, Prentice Hall, Englewood Cliffs, NJ. (1992).  
BROWN, D., PORTA, H. i UHL, J., *Calculus & Mathematica.*, Addison-Wesley, Reedwood City, CA. (1991).  
CHRISTENSEN, S. M., Resources for Computer Algebra., *Computers in Physics*, **8**, 308–316 (1994).  
GEDDES, K. O. *et al.*, *Algorithms for Computer Algebra.*, Kluwer Academic Publishers, Boston (1992).  
HECK, A., *Introduction to Maple.*, Springer-Verlag, New York (1993).  
RAYNA, G., *REDUCE: Software for Algebraic Computation.*, Springer-Verlag, New York (1987).



Plasma d'inducció acoblada. Espectròmetres multicanal i seqüencial. (Serveis Científico-tècnics de la UB)