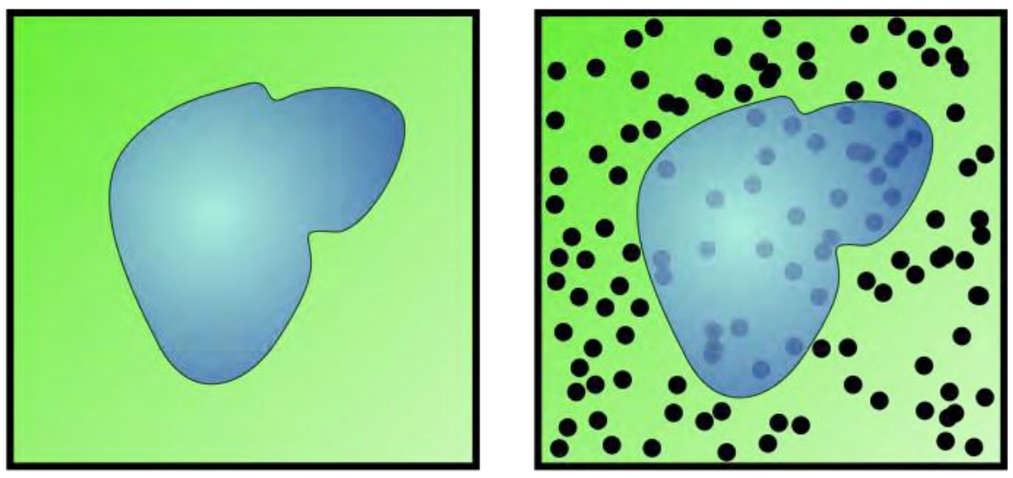
Noite Europeia dos Investigadores

Dez Algoritmos do Séc. XX

Método Monte-Carlo

[John von Neumann, Stan Ulam e Nick Metropolis | 1946]



Estes métodos usam números gerados aleatoriamente (isto é, à sorte). Neste exemplo, para calcular aproximadamente a área da superfície do lago que está dentro de um terreno quadrado de área A, podemos atirar pedras ao acaso para dentro do quadrado: por exemplo, se um terço das pedras cair no lago, aproximamos a área do lago por A/3. Numa simulação computacional, "atirar a pedra" consiste em gerar, de forma aleatória, as coordenadas planas (x,y) do ponto onde "cai a pedra".

Método *simplex*

[George Dantzig | 1947]

A programação linear (também designada por otimização linear) é omnipresente em todas as atividades humanas onde o sucesso económico depende da capacidade de obter bons resultados com economia de custos. O problema real é modelado por relações matemáticas entre as variáveis envolvidas e o "*simplex*" é um método eficiente para produzir a solução que joga um papel fundamental na tomada de decisões (por exemplo, pelo gestor de uma empresa).

Métodos de Krylov

[Magnus Hestenes, Eduard Stiefel e Cornelius Lanczos | 1950]

Resolver um sistema com duas equações lineares e duas incógnitas, por exemplo o sistema x +y=1 e 2x+y=3, é fácil. O caso muda radicalmente de figura se se tratar de um sistema com n equações e n incógnitas, com n muito grande, digamos dez mil. Em muitas aplicações, cada equação envolve não todas mas apenas algumas incógnitas e torna-se vantajoso recorrer a métodos iterativos, entre os quais os métodos de Krylov ocupam lugar de destaque.

Decomposição de matrizes

[Alston Householder | 1951]

A possibilidade de expressar uma matriz na forma de um produto de outras duas (ou mais) matrizes com forma especial (triangular, diagonal, ortogonal, etc.) permitiu o desenvolvimento de bibliotecas da álgebra linear numérica (que são o núcleo computacional de muita da computação científica) flexíveis, robustas e eficientes.

Compilador Fortran

[John Bacus | 1957]

Apesar de não ser historicamente a primeira linguagem de programação de alto nível, o Fortran foi a primeira a ter grande uso. Com uma linguagem de alto nível, os utilizadores podem "dizer" ao computador o que fazer sem ter que descer ao mundo intrincado do código-máquina.

Algoritmo QR

[J.G.F. Francis | 1959-61]

O cálculo de valores e vetores próprios de matrizes tem muitas aplicações. Por exemplo, uma boa parte do sucesso do motor de busca Google reside num algoritmo (designado por *PageRank*) que se baseia no cálculo de um certo vetor próprio da maior matriz com que se trabalha nas aplicações dos dias de hoje, a matriz do Google. O algoritmo QR, embora outros algoritmos para calcular valores e vetores próprios tenham aparecido depois dele, é ainda hoje muito popular na comunidade científica (mas não é o algoritmo usado no Google).

Quicksort

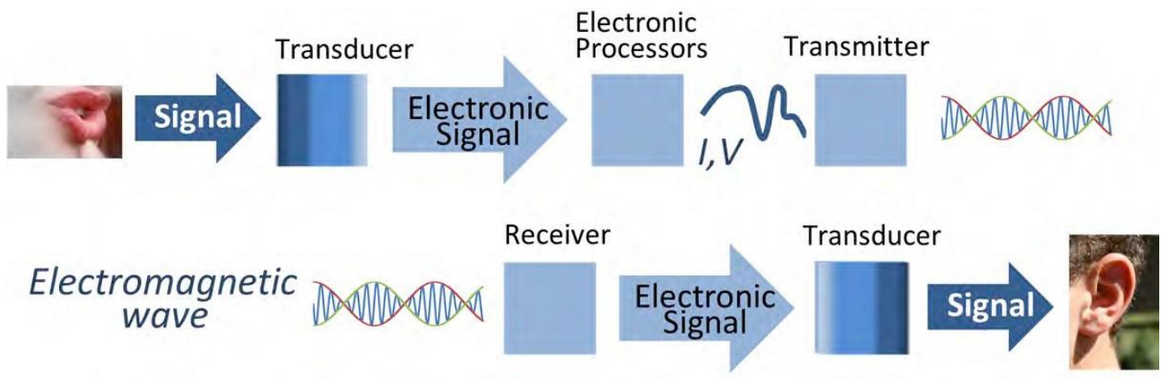
[Tony Hoare | 1962]

Colocar n nomes de uma dada lista por ordem alfabética ou n números por ordem numérica é uma tarefa frequente. Existem vários algoritmos para este fim. O *Quicksort* é um algoritmo que usa uma estratégia de "dividir-para-reinar". Pode acontecer que este algoritmo precise de n(n-1)/2 comparações, mas em média são necessárias muito menos (um número da ordem de n.log(n))

Transformada rápida de Fourier

[James Coley e John Tukey | 1965]

Considerado por muitos como o algoritmo com mais aplicações, ele revolucionou a área do processamento de sinal por reduzir o custo da transformação de N sinais de O(N^2) para O(N,log(N)). A Transformada rápida de Fourier é usada em *transducers* que, por exemplo, convertem sinais sonoros em sinais eletrónicos e vice-versa.



Deteção de relações inteiras entre números reais

[Helaman Ferguson e Rodney Forcade | 1977]

Este é um problema antigo mas com aplicações modernas (por exemplo, na criptografia). Dados números reais x1, x2, …, xn, o algoritmo encontra, se existirem, números inteiros a1, a2, …, an (não todos nulos) tais que a1*x1+a2*x2+…+an*xn=0.

Método rápido dos multipolos

[Leslie Greengard e Vladimir Rokhlin | 1987]

No cálculo do movimento de n partículas em que cada uma interfere com as restantes através de forças gravitacionais ou electroestáticas (por exemplo as estrelas de uma galáxia ou os átomos de uma proteína) parece inevitável que os algoritmos tenham complexidade O(n^2), uma vez que com n partículas temos n(n-1)/2 pares distintos. Mas o algoritmo dos multipolos reduz a complexidade para O(n), aproximando os efeitos de um grupo distante de partículas sobre um grupo local.

Universidade do Minho
Escola de Ciências
Departamento de Matemática e Aplicações