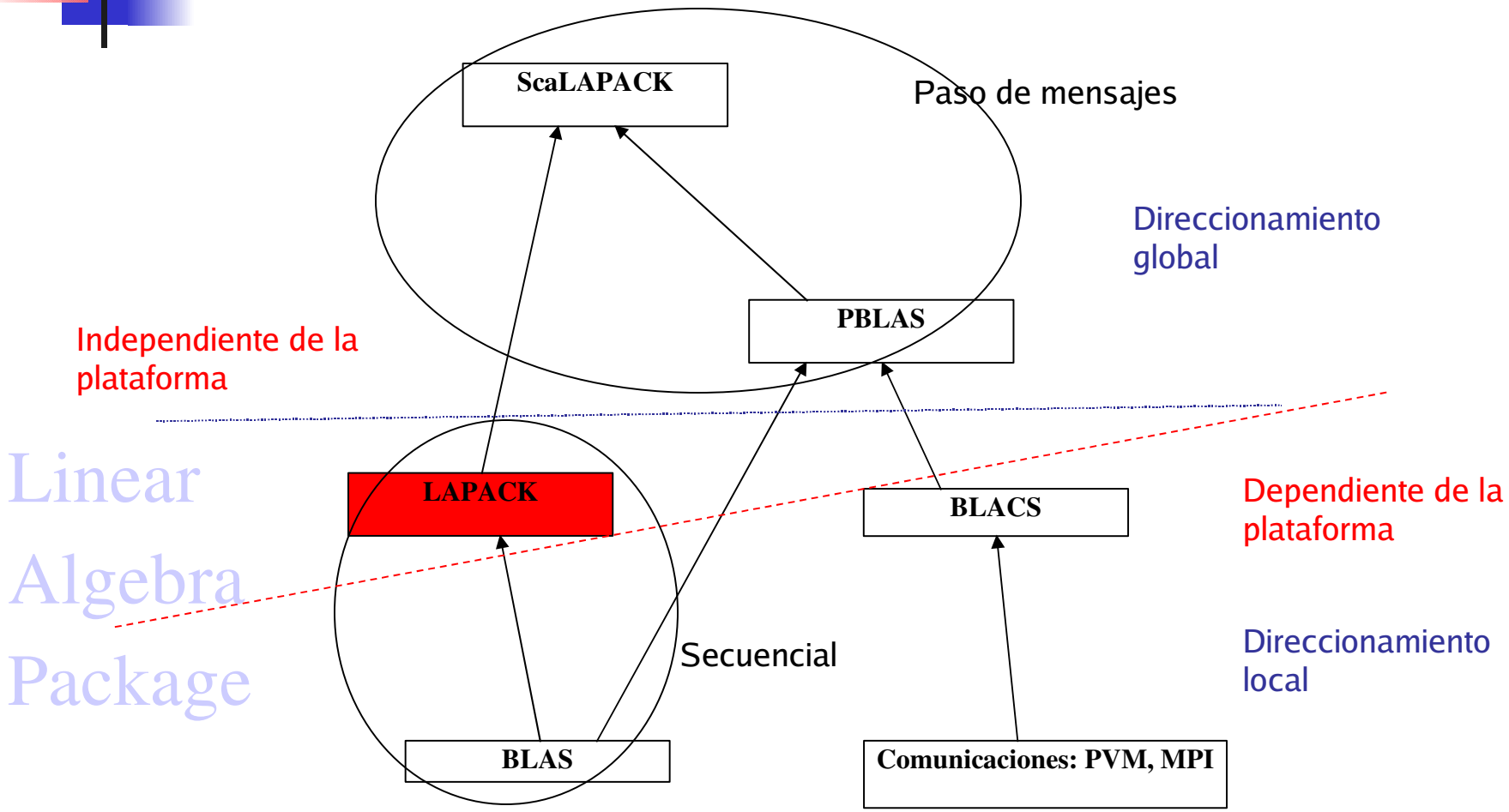
Librería secuencial de Álgebra Lineal Densa LAPACK

Domingo Giménez
Javier Cuenca

Facultad de Informática
Universidad de Murcia

LAPACK

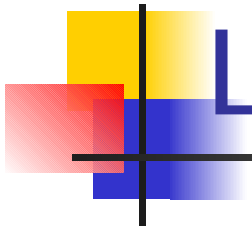

ScaLAPACK
Paso de mensajes
Direccionamiento global
PBLAS
Independiente de la plataforma
Linear Algebra Package
LAPACK
BLACS
Dependiente de la plataforma
Secuencial
Direccionamiento local
BLAS
Comunicaciones: PVM, MPI

LAPACK

- Conjunto de rutinas para resolver problemas de los más frecuentes en álgebra lineal densa: sistemas de ecuaciones y problemas de valores propios
  - **Documentos:**

Implementation Guide for LAPACK
UT, CS-90-101, April 1990.
E. Anderson and J. Dongarra

LAPACK: A Portable Linear Algebra Library for High-Performance Computers
UT, CS-90-105, May 1990.
E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen



LAPACK

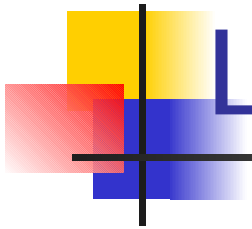
- Algoritmos orientados a bloques

$\Rightarrow$

Basados en BLAS

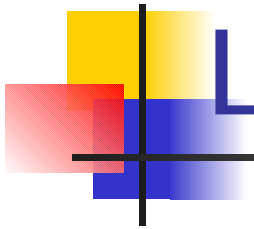
Eficiencia

Portabilidad



LAPACK

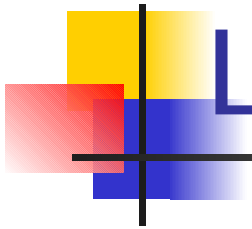
- Problemas que resuelve:
  - Sistemas de ecuaciones lineales
  - Problemas de mínimos cuadrados
  - Problemas de valores propios
  - Problemas de valores singulares
  - Otros: factorización de matrices, estimación del número de condición, etc.



LAPACK

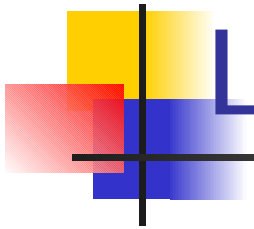
- Tipos de matrices:
  - Densas.
  - Banda.
  - Reales y complejas.

    … no escasas
- Tipos de sistemas:
  - Secuenciales.
  - Memoria compartida.



LAPACK

- Tipos de rutinas:
  - “Driver routines” – Rutinas conductoras.
    - Resuelve un problema.
  - “Computational routines” – Rutinas computacionales.
    - Realizan una tarea computacional
  - “Auxiliary routines” – Rutinas auxiliares.
    - Realizan una subtarea o trabajo de menor nivel.

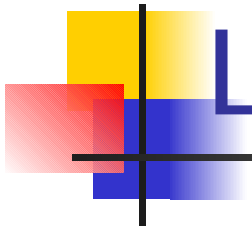


LAPACK. Tipos de rutinas

- **Rutinas conductoras**:
  - Para la resolución completa de problemas estándar:
    - Sistemas de ecuaciones lineales.
    - Problemas de valores propios.
  - Siempre que sea posible es recomendable usar estas rutinas para resolver un problema.

LAPACK. Tipos de rutinas

- **Rutinas computacionales**:
  - Realizan tareas computacionales:
    - Factorizaciones LU y QR, reducción de matriz simétrica a tridiagonal, ...
  - Cada rutina conductora realiza una secuencia de llamadas a las rutinas computacionales.
  - El usuario también puede llamar en sus programas a rutinas computacionales.



LAPACK. Tipos de rutinas

- **Rutinas auxiliares**:
  - Son rutinas que hacen operaciones de bajo nivel:
    - Versiones no orientadas a bloques de algoritmos orientados a bloques.
    - Computaciones de bajo nivel (escalar una matriz, generación de matriz de Householder).
    - Extensiones de BLAS.

LAPACK

- Formato de rutinas conductoras y computacionales: XYYZZZ

X: Tipo de datos:

S : REAL
D : DOUBLE PRECISION
C : COMPLEX
Z : DOUBLE COMPLEX

YY: Tipo de matriz

ZZZ: Operación:

SV: sistemas de ecuaciones
EV: valores propios ...

LAPACK

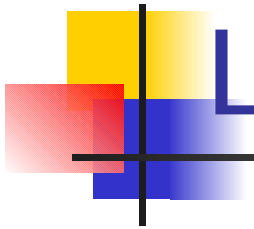
Tipos de matrices YY (1/2):

BD bidiagonal;
GB general band;
GE general (i.e., unsymmetric, in some cases rectangular);
GG general matrices, generalized problem (i.e., a pair of general matrices);
GT general tridiagonal;
HB (complex) Hermitian band;
HE (complex) Hermitian;
HG upper Hessenberg matrix, generalized problem (i.e a Hessenberg and a triangular matrix);
HP (complex) Hermitian, packed storage;
HS upper Hessenberg;
OP (real) orthogonal, packed storage;
OR (real) orthogonal;
PB symmetric or Hermitian positive definite band;
PO symmetric or Hermitian positive definite;
PP symmetric or Hermitian positive definite, packed storage;

LAPACK

- Tipos de matrices YY (2/2):

```
PT symmetric or Hermitian positive definite tridiagonal;
SB (real) symmetric band;
SP symmetric, packed storage;
ST (real) symmetric tridiagonal;
SY symmetric;
TB triangular band;
TG triangular matrices, generalized problem (i.e., a
pair of triangular matrices);
TP triangular, packed storage;
TR triangular (or in some cases quasi-triangular);
TZ trapezoidal;
UN (complex) unitary;
UP (complex) unitary, packed storage
```



LAPACK

- Rutinas conductoras de resolución de ecuaciones lineales: AX = B
  - Rutina simple: xyySV
    - Factoriza A y sobreescribe B con X
  - Rutina experta: xyySVX. Puede llevar a cabo otras funciones:
    - $A^TX=B$ o $A^HX=B$
    - Número de condición, singularidad, ...
    - Refina la solución y hace análisis de error.
    - Equilibrado del sistema.

LAPACK. Ejemplo dgesv

- Ejemplo dgesv
  - Resuelve un sistema de ecuaciones
  - Llamada en Fortran:

    call dgesv( )
  - En C:

    dgesv_( )

    y se pasan las referencias a los parámetros

LAPACK. Ejemplo dgesv

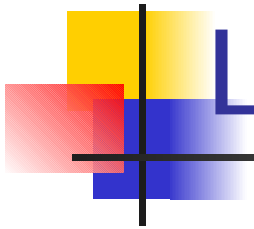
- dgesv
  - Rutina conductora de LAPACK
  - Resolución de un sistema de ecuaciones **AX=B**
  - Llamadas:
    - dgetrf
      - Rutina computacional de LAPACK
      - Factorización LU: Transforma **A → LU**
    - dgetrs
      - Rutina computacional de LAPACK
      - Resuelve el doble sistema triangular **LU X = B**

LAPACK. Ejemplo dgesv

- dgetrf
  - Rutina computacional de LAPACK
  - Factorización LU: Transforma **A → LU**
  - Llamadas en cada pasada de bucle:
    - dgetf2
      - Rutina auxiliar de LAPACK
      - Factorización LU sin bloques aplicada a determinados bloques de A
    - dtrsm (2 veces por pasada)
      - Rutina del nivel 3 de BLAS
      - Resuelve un sistema triangular de ecuaciones
    - dgemm
      - Rutina del nivel 3 de BLAS
      - Multiplicación de matrices

LAPACK. Ejemplo dgesv

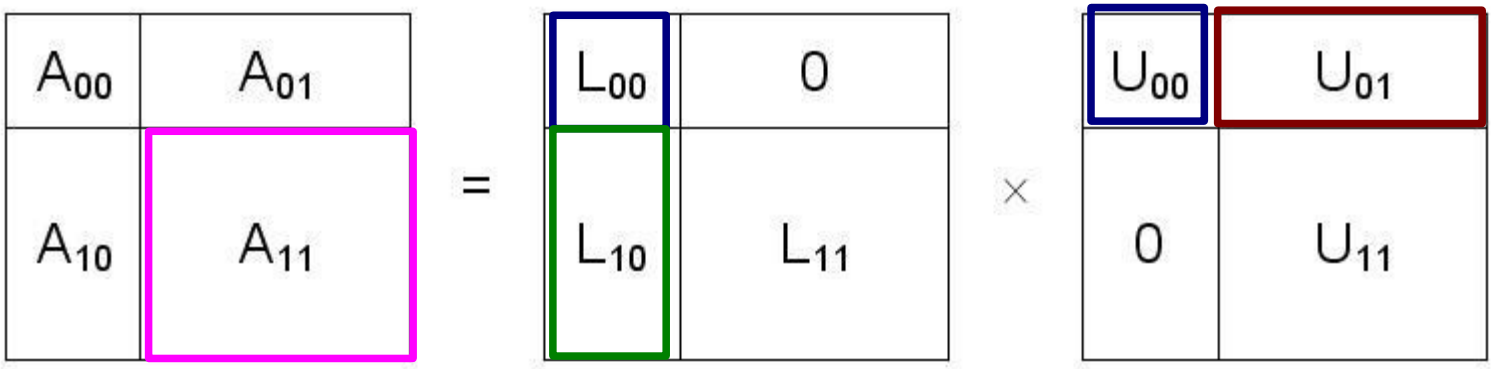
- dgetrs
  - Rutina computacional de LAPACK
  - Resuelve el doble sistema triangular **LU X =B**
  - Llamadas en cada pasada de bucle:
    - dlaswp
      - Rutina auxiliar de LAPACK
      - Aplica a **B** los intercambios de filas realizados previamente a las matrices **L** y **U**
    - dtrsm
      - Rutina del nivel 3 de BLAS
      - Resuelve un sistema triangular de ecuaciones **LY=B**
    - dtrsm
      - Rutina del nivel 3 de BLAS
      - Resuelve un sistema triangular de ecuaciones **UX=Y**



LAPACK

- También:
  - Valores propios no simétrico.
  - Descomposición en valores singulares.
  - Valores propios simétrico generalizado.
  - Valores propios no simétrico generalizado.
  - Descomposición en valores singulares generalizado.

Factorización LU



Cada A_{ij}, L_{ij}, U_{ij} de tamaño b b

Paso 1: $L_{00} U_{00} = A_{00} \Rightarrow$ Factorización sin bloques

Paso 2: $L_{00} U_{01} = A_{01} \Rightarrow$ Sistema múltiple triangular inferior (¿bloques?)

Paso 3: $L_{10} U_{00} = A_{10} \Rightarrow$ Sistema múltiple triangular superior (¿bloques?)

Paso 4: $A_{11} = L_{10} U_{01} + L_{11} U_{11} \Rightarrow A'_{11} = A_{11} - L_{10} U_{01}$, por bloques

y seguir trabajando con el nuevo valor de A_{11}

Factorización LU

```
void lu_bloques(double *a,int fa,int ca,int lda,int tb)
{int i,j,k,f,c;
  for(i=0;i<fa;i=i+tb)  {
    f=(tb<fa-i ? tb : fa-i);    c=(tb<ca-i ? tb : ca-i);
    lu(&a[i*lda+i],f,c,lda);
    if(i+tb<fa)    {
      sistema_triangular_inferior(&a[i*lda+i],f,c,lda,&a[i*lda+i+c],f,ca-i-c,lda);
     sistema_triangular_superior(&a[i*lda+i],f,c,lda,&a[(i+f)*lda+i],fa-i-f,c,lda);
      multiplicar_restar_matrices(&a[(i+f)*lda+i],fa-i-f,c,lda,
        &a[i*lda+i+f],f,ca-i-c,lda,&a[(i+f)*lda+i+c],fa-i-f,ca-i-c,lda);
    } } }
```

Factorización LU

en mi portátil:

tamaño bloque\matriz	800	1000
1	2.10	4.01
12	1.42	2.78
25	1.29	2.27
37	1.24	2.37
44	1.20	2.00
50	1.22	2.32
100	1.47	2.24
200	2.29	3.47
400	2.17	3.67
sin bloques	1.73	3.43