

# Capacitación ScaLAPACK: Simulación de yacimientos mediante descomposición matricial

Oscar Peredo A.

Centro de Modelamiento Matemático,  
Laboratorio ALGES,  
Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile

3 de Agosto de 2010

# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# Esquema

---

## Introducción

## Objetivos

## Método de descomposición matricial

## ScaLAPACK

## Rutina PDPOTRF

## Utilización desde lenguaje C



# Motivación

---



# Motivación

---

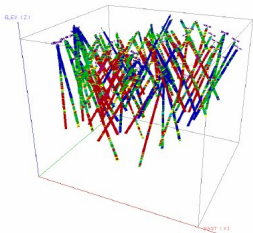
En evaluación de yacimientos se busca estimar los recursos existentes a partir de muestras de sondajes.



# Motivación

---

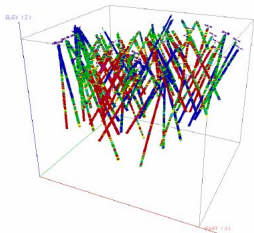
En evaluación de yacimientos se busca estimar los recursos existentes a partir de muestras de sondajes.



# Motivación

---

En evaluación de yacimientos se busca estimar los recursos existentes a partir de muestras de sondajes.



Típicamente esa estimación se realiza utilizando un método llamado *kriging* ([Kri51, Mat62, Mat63]), el cual interpola valores entre los nodos de una grilla utilizando los datos conocidos.



# Motivación

---





# Motivación

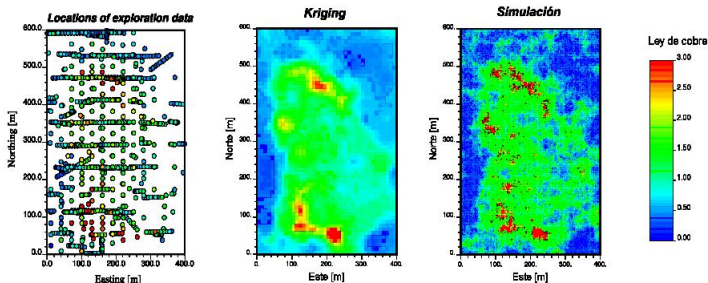
---

Además de la estimación, se hace simulación para determinar la variabilidad de las leyes (y distribución espacial).



# Motivación

Además de la estimación, se hace simulación para determinar la variabilidad de las leyes (y distribución espacial).



Datos 2D extraídos del sondaje (izquierda), estimación de los datos mediante kriging (centro), simulación de los datos (derecha).



# Motivación

---



# Motivación

---

Los métodos de simulación geoestadística más conocidos son los siguientes ([Goo97, Deu02]):



# Motivación

---

Los métodos de simulación geoestadística más conocidos son los siguientes ([Goo97, Deu02]):

- Descomposición matricial
- Espectral
- Bandas rotantes
- Secuencial gaussiano
- Secuencial indicador
- Gaussiano truncado
- Recocido simulado



# Motivación

---

Los métodos de simulación geoestadística más conocidos son los siguientes ([Goo97, Deu02]):

- **Descomposición matricial** [Dav87]
- Espectral
- Bandas rotantes
- Secuencial gaussiano
- Secuencial indicador
- Gaussiano truncado
- Recocido simulado



# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# Objetivos

---





# Objetivos

---

- Revisión del método de descomposición matricial.



# Objetivos

---

- Revisión del método de descomposición matricial.
- Descripción y estructura de la librería ScaLAPACK.



# Objetivos

---

- Revisión del método de descomposición matricial.
- Descripción y estructura de la librería ScaLAPACK.
- Utilización de la librería desde un programa en lenguaje C.



# Objetivos

---

- Revisión del método de descomposición matricial.
- Descripción y estructura de la librería ScaLAPACK.
- Utilización de la librería desde un programa en lenguaje C.
- Utilización de la rutina PDPOTRF, la cual realiza la descomposición de Cholesky de una matriz simétrica definida positiva en un sistema de memoria distribuída.



# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# Método de descomposición matricial

---



# Método de descomposición matricial

---

- Este método utiliza como base una estimación ya calculada de la imagen a simular (*kriging*).



# Método de descomposición matricial

---

- Este método utiliza como base una estimación ya calculada de la imagen a simular (*kriging*).
- Posteriormente se particionan los datos en 2 grupos, los nodos a simular (estimados mediante *kriging*) y los datos condicionantes (datos conocidos mediante sondeo).





# Método de descomposición matricial: Algoritmo

---



# Método de descomposición matricial: Algoritmo

- Construir matriz de covarianzas espaciales entre los nodos a simular y los datos condicionantes:

$$C = \left[ \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right]$$

donde  $C_{11}$  son las covarianzas entre nodos a simular y nodos a simular,  $C_{22}$  entre puntos con datos condicionantes y puntos con datos condicionantes, y  $C_{12}$  entre nodos a simular y puntos con datos.

- Calcular descomposición de Cholesky de la matriz  $C = L \cdot L^T$ .
- Generar un vector aleatorio:

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

donde  $w_1$  es un vector aleatorio generado con distribución  $N(0, 1)$ , y  $w_2$  son los datos condicionantes normalizados.

- Obtener valores simulados  $y_{sim} = L \cdot w$ , los cuales reproducen la covarianza  $C$ .



# Método de descomposición matricial: Algoritmo

- Construir matriz de covarianzas espaciales entre los nodos a simular y los datos condicionantes:

$$C = \left[ \begin{array}{c|c} C_{11} & C_{12} \\ \hline C_{21} & C_{22} \end{array} \right]$$

donde  $C_{11}$  son las covarianzas entre nodos a simular y nodos a simular,  $C_{22}$  entre puntos con datos condicionantes y puntos con datos condicionantes, y  $C_{12}$  entre nodos a simular y puntos con datos.

- Calcular descomposición de Cholesky de la matriz  $C = L \cdot L^T$ .
- Generar un vector aleatorio:

$$w = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

donde  $w_1$  es un vector aleatorio generado con distribución  $N(0, 1)$ , y  $W_2$  son los datos condicionantes normalizados.

- Obtener valores simulados  $y_{sim} = L \cdot w$ , los cuales reproducen la covarianza  $C$ .



# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# ScaLAPACK: estructura

---



# ScaLAPACK: estructura

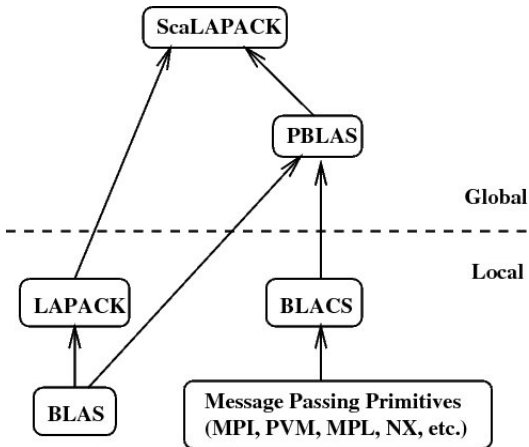
---

*Scalable Linear Algebra PACKage* [BCC<sup>+</sup>96]



# ScaLAPACK: estructura

*Scalable Linear Algebra PACKage* [BCC<sup>+</sup>96]



# ScaLAPACK: distribución 2D cíclica por bloques de filas

---





# ScaLAPACK: distribución 2D cíclica por bloques de filas

---

- Tamaño de los bloques:  $r \times c$ .



# ScaLAPACK: distribución 2D cíclica por bloques de filas

---

- Tamaño de los bloques:  $r \times c$ .
- $P$  procesos se distribuyen en una grilla de  $P_r \times P_c$  celdas.



# ScaLAPACK: distribución 2D cíclica por bloques de filas

---

- Tamaño de los bloques:  $r \times c$ .
- $P$  procesos se distribuyen en una grilla de  $P_r \times P_c$  celdas.
- El elemento  $(m, n)$ -ésimo de una matriz se almacenará en la  $(i, j)$ -ésima celda del bloque  $(b, d)$  a través del proceso  $(p, q)$  donde:

$$\langle (p, q), (b, d), (i, j) \rangle := \langle (\lfloor \frac{m}{r} \rfloor \bmod P_r, \lfloor \frac{n}{c} \rfloor \bmod P_c),$$

$$\left( \left\lfloor \frac{\lfloor \frac{m}{r} \rfloor}{P_r} \right\rfloor, \left\lfloor \frac{\lfloor \frac{n}{c} \rfloor}{P_c} \right\rfloor \right),$$

$$(m \bmod r, n \bmod c) \rangle$$



# ScaLAPACK: distribución 2D cíclica por bloques de filas

---



# ScaLAPACK: distribución 2D cíclica por bloques de filas

Ejemplo:  $P_r = P_c = r = c = 2$ , matriz de  $5 \times 5$ .

$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$
$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$a_{35}$
$a_{41}$	$a_{42}$	$a_{43}$	$a_{44}$	$a_{45}$
$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55}$

	0	1
0	$a_{11}$ $a_{12}$ $a_{15}$ $a_{21}$ $a_{22}$ $a_{25}$ $a_{51}$ $a_{52}$ $a_{55}$	$a_{13}$ $a_{14}$ $a_{23}$ $a_{24}$ $a_{53}$ $a_{54}$
1	$a_{31}$ $a_{32}$ $a_{35}$ $a_{41}$ $a_{42}$ $a_{45}$	$a_{33}$ $a_{34}$ $a_{43}$ $a_{44}$



# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# Rutina PDPOTRF

---



# Rutina PDPOTRF

---

- Realiza la descomposición de Cholesky en paralelo para sistemas de memoria distribuída.





# Rutina PDPOTRF

---

- Realiza la descomposición de Cholesky en paralelo para sistemas de memoria distribuída.
- Implementa la factorización *outer-cholesky* [GVL96]:
  - 1: **Para**  $k = 1, \dots, n$  **hacer**
  - 2:    $A(k, k) \leftarrow \sqrt{A(k, k)}$
  - 3:    $A((k + 1) : n, k) \leftarrow \frac{1}{A(k, k)} A((k + 1) : n, k)$
  - 4:   **Si**  $j = k + 1, \dots, n$  **entonces**
  - 5:      $A(j : n, j) \leftarrow A(j : n, j) - A(j : n, k)A(j, k)$
  - 6:   **Fin (Si)**
  - 7: **Fin (Para)**



# Esquema

---

Introducción

Objetivos

Método de descomposición matricial

ScaLAPACK

Rutina PDPOTRF

Utilización desde lenguaje C



# Utilización desde lenguaje C

---



# Utilización desde lenguaje C

---

- Enlaces dinámicos o estáticos a las librerías asociadas a ScaLAPACK: BLACS, BLAS, LAPACK.



# Utilización desde lenguaje C

---

- Enlaces dinámicos o estáticos a las librerías asociadas a ScaLAPACK: BLACS, BLAS, LAPACK.
- Diversas implementaciones de las librerías anteriores: ATLAS, AMD ACML, Apple Velocity Engine, Cray libsci, Intel MKL, etc.



# Utilización desde lenguaje C

---

- Enlaces dinámicos o estáticos a las librerías asociadas a ScaLAPACK: BLACS, BLAS, LAPACK.
- Diversas implementaciones de las librerías anteriores: ATLAS, AMD ACML, Apple Velocity Engine, Cray libsci, **Intel MKL**, etc.
- Definir `include` (dependen de las librerías enlazadas).



# Utilización desde lenguaje C

---

- Enlaces dinámicos o estáticos a las librerías asociadas a ScaLAPACK: BLACS, BLAS, LAPACK.
- Diversas implementaciones de las librerías anteriores: ATLAS, AMD ACML, Apple Velocity Engine, Cray libsci, **Intel MKL**, etc.
- Definir `include` (dependen de las librerías enlazadas).
- Utilización de un `Makefile`.



# Utilización desde lenguaje C: enlaces

<http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/>

Select OS:	Linux*
Select processor architecture:	Intel(R) 64
Select compiler:	GNU C
Select dynamic or static linking:	Static
Select your integers length:	32-bit (lp64)
Select sequential or multi-threaded version of Intel® MKL:	Sequential
Select OpenMP library:	<Select OpenMP>
Select cluster library:	ScaLAPACK
Select MPI library:	Open MPI

Use this link line:

```
$MKLPATH/libmkl_solver_lp64_sequential.a -Wl,--start-group  
$MKLPATH/libmkl_intel_lp64.a $MKLPATH/libmkl_sequential.a  
$MKLPATH/libmkl_core.a $MKLPATH/libmkl_blacs_openmpi_lp64.a -Wl,--end-group -lpthread
```

\* The MKLPATH environment variable should point to the directory containing the Intel MKL libraries.

\*\* You may remove libraries with 'solver' in the name from the link line if you do not use the iterative solver or multiprecision integer functionality.





# Utilización desde lenguaje C: include

---

```
#include "mpi.h"
#include "PBblacs.h"

extern int descinit_(int *desc, int *m, int *n, int *mb,
                    int *nb, int *irsrc, int *icsrc,
                    int *ictxt, int *lld, int *info);
extern int numroc_( int *n, int *nb, int *iproc,
                  int *isrcproc, int *nprocs);
extern int pdelset_( double *a, int *ia, int *ja,
                  int *desca, double *alpha);
extern int pdpotrf_( char *uplo, int *n, double *a,
                  int *ia, int *ja, int *desca,
                  int *info);
extern void pdgemr2d_(int *m, int *n, double *a, int *ia,
                    int *ja, int *desc_a, double *b,
                    int *ib, int *jb, int *desc_b,
                    int *ictxt);
```



# Utilización desde lenguaje C: Makefile

---

```
CC=mpicc
CFLAGS= -Wall -g -O0 -I../include -DBIT64
FFLAGS=

MKLLIB=/opt/intel/Compiler/11.0/083/mkl/lib/em64t

SCALAPACKLIB= $(MKLLIB)/libmkl_scalapack_lp64.a \
              $(MKLLIB)/libmkl_solver_lp64_sequential.a \
              -Wl,--start-group \
              $(MKLLIB)/libmkl_intel_lp64.a \
              $(MKLLIB)/libmkl_sequential.a \
              $(MKLLIB)/libmkl_core.a \
              $(MKLLIB)/libmkl_blacs_openmpi_lp64.a \
              -Wl,--end-group \
              -lpthread

LIBLM= -lm

pdpotrf2: pdpotrf2.o
$(CC) $(CFLAGS) pdpotrf2.o $(SCALAPACKLIB) $(LIBLM) -o pdpotrf2
rm -f *.o

clean:
rm pdpotrf2
rm -f *.o*
rm -f *.e*
rm -f *.po*
rm -f *.pe*
```



# Utilización desde lenguaje C: código

---

Ir al código...



# Utilización desde lenguaje C: *speedup*

---



# Utilización desde lenguaje C: *speedup*

---

Cluster leloo (ALGES): 4 nodos, 8 cores c/u, 16GB RAM, IB.



## Utilización desde lenguaje C: *speedup*

---

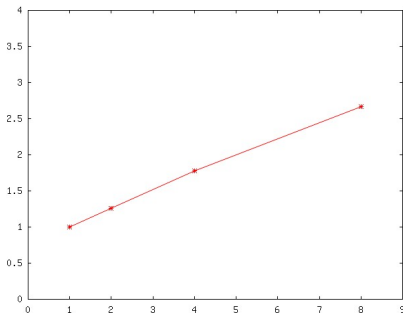
Cluster leloo (ALGES): 4 nodos, 8 cores c/u, 16GB RAM, IB.  
Speedup (cálculo de descomposición de Cholesky para matriz simétrica definida positiva de rango 10176):



## Utilización desde lenguaje C: *speedup*

Cluster leloo (ALGES): 4 nodos, 8 cores c/u, 16GB RAM, IB.  
Speedup (cálculo de descomposición de Cholesky para matriz simétrica definida positiva de rango 10176):

	1 p	2 p	4 p	8 p
tiempo (s)	439.15	349.94	245.68	164.08
<i>speedup</i> ( $t_S/t_P$ )	1	1.26	1.78	2.67



# Enlaces

---

- [http://www.dim.uchile.cl/ operedo/scalapack/](http://www.dim.uchile.cl/operedo/scalapack/)
- <http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor/>
- 





# Bibliografía

---

- 
- L. S. Blackford, J. Choi, A. Cleary, A. Petitet, R. C. Whaley, J. Demmel, I. Dhillon, K. Stanley, J. Dongarra, S. Hammarling, G. Henry, and D. Walker, *Scalapack: a portable linear algebra library for distributed memory computers - design issues and performance*, Supercomputing '96: Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM) (Washington, DC, USA), IEEE Computer Society, 1996, p. 5.
- 
- M. Davis, *Production of conditional simulations via the lu triangular decomposition of the covariance matrix*, Mathematical Geology **19** (1987), no. 2, 91–98.
- 
- C. V. Deutsch, *Geostatistical reservoir modeling*, Oxford University Press, 2002.
- 
- P. Goovaerts, *Geostatistics for natural resources evaluation*, Oxford University Press New York, 1997.
- 
- G. H. Golub and C. F. Van Loan (eds.), *Matrix computations, third edition*, The Johns Hopkins University Press, Baltimore, 1996.
- 
- D. G. Krige, *A statistical approach to some mine valuations and allied problems at the witwatersrand*, Master's thesis, University of Witwatersrand (1951).
- 
- G. Matheron, *Traite de geostatistique appliquee, tome i*, Memoires du Bureau de Recherches Geologiques et Minieres **14** (1962).
- 
- , *Traite de geostatistique appliquee, tome ii*, Editions Techniques, Paris (1963).

