

PyACTS Documentation:

PyBLACS, PyPBLAS and PyScaLAPACK Quick Reference

Tony Drummond

LADrummond@lbl.gov

Lawrence Berkeley National Laboratory

One Cyclotron Road, Berkeley CA 94703, U.S.A.

Vicente Galiano

vgaliano@umh.es

Dpto. Física y Arquitectura de Computadores

U.M.H. 03202 Elche, Alicante, Spain

Violeta Migallón, José Penadés

{violeta, jpenades}@dccia.ua.es

Dpto. Ciencia Computación e Inteligencia Artificial

Universidad de Alicante, E-03071 Alicante, Spain

PyACTS Tools Quick Reference

INICIALIZACIÓN Y LIBERACIÓN

```
gridinit([file_config='', mb, nb, nprow, npcol])
gridexit()
```

Inicializa la malla de procesos

Libera todos los recursos asignados a una malla de procesos

VERIFICACIÓN DE VARIABLES PyACTS

```
IsACTSlib(ACTS_var)
readACTSdesc(ACTSArray)
desc=ACTSgetdesc(a[, ACTS_lib, ia, ja])
desc=ACTSmakedesc(m, n[, ACTS_lib, ia, ja])
m, n=getdims(x)
```

Comprueba si una variable es un array PyACTS

Devuelve el descriptor de un array PyACTS

Devuelve el descriptor desc de la matriz global a de tipo Numeric

obtiene un descriptor a partir del tamaño de la matriz (m x n)

Devuelve las dimensiones de una matriz local en todos los procesos

CONVERSIÓN

```
a_acts=Num2PyACTS(a, ACTS_lib)
a=PyACTS2Num(a_acts)
a_acts=Rand2PyACTS(m, n, ACTS_lib)
alpha_acts=Scal2PyACTS(alpha, ACTS_lib)
a_acts=PNetCDF2PyACTS(a_pnetcdf, ACTS_lib)
a_netcdf=PyACTS2PNetCDF(a_acts, a_netcdf)
```

Convierte una variable Numeric y local en una variable PyACTS distribuida

Convierte una variable PyACTS distribuida en una variable Numeric local en el nodo 0

Crea una variable PyACTS con las dimensiones indicadas con datos aleatorios

Distribuye un escalar todos los elementos de la malla

Lee de forma distribuida un fichero NetCDF a formato PyACTS

Escribe de forma distribuida de una variable PyACTS a un fichero NetCDF

PyBLACS Quick Reference

INICIALIZACIÓN <code>iam,nprocs=blacspinfo()</code> <code>iam,nprocs = setup(nprocs)</code> <code>val = get(what,[ictxt])</code> <code>ictxt = gridinit(nprow,npcol[,order])</code> <code>ictxt = gridmap(ictxt,usermap,ldumap, nprow,npcol)</code>	 Obtiene identidad y número de procesos (MPI) Obtiene identidad y número de procesos (PVM) Obtiene valores internos de las BLACS Inicializa la malla con tamaño indicado Establece cada proceso dentro de una malla
DESTRUCCIÓN <code>abort([ictxt,errornum])</code> <code>gridexit(ictxt)</code> <code>exit([ictxt])</code> <code>freebuff([ictxt,wait])</code>	 Mata todos los procesos BLACS cuando ha habido algun error Libera los recursos del contexto <code>ictxt</code> Libera TODOS los recursos de todos los contextos Libera el buffer de las BLACS
INFORMATIVAS <code>nprow,npcol,myrow,mycol=gridinfo([ictxt])</code> <code>prow,pcol = pcoord(pnum[,ictxt])</code> <code>nprow,npcol,myrow,mycol=gridinfo([ictxt])</code> <code>num = pnum(prow,pcol,[ictxt])</code>	 Proporciona información de la malla con contexto <code>ictxt</code> Devuelve las coordenadas de un identificador dentro de la malla <code>ictxt</code> Proporciona información de la malla con contexto <code>ictxt</code> Proporciona el identificador a partir de las coordenadas de un proceso de la malla con contexto <code>ictxt</code>
ENVÍO <code>gesd2d(a,rdest,cdest,[ictxt,llda])</code> <code>trsd2d(a,rdest,cdest,[ictxt,llda])</code>	 Envia los datos de la matriz a hacia (<code>rdest,cdest</code>) Envia los datos de la matriz triangular a hacia (<code>rdest,cdest</code>)
RECEPCIÓN <code>a=gerv2d(a,rsrc,csrc,[ictxt,llda])</code> <code>a=trrv2d(a,rsrc,csrc,[ictxt,llda,diag,llda])</code>	 Recibe los datos de la matriz a desde (<code>rsrc,csrc</code>) Recibe los datos de la matriz triangular a desde (<code>rsrc,csrc</code>)
DIFUSIÓN <code>gebs2d(a,[ictxt,scope,top,llda])</code> <code>trbs2d(a[,ictxt,scope,top,uplo,diag,llda])</code> <code>a=gebr2d(a,irsrc,icsrc[,ictxt,scope,top,lda])</code> <code>a=trbr2d(a,irsrc,icsrc[,ictxt,uplo,diag,scope,top,lda])</code>	 Inicia la difusión al grupo de destinatarios indicados por <code>scope</code> Inicia la difusión de una matriz triangular al grupo de destinatarios indicados por <code>scope</code> Recibe los datos provenientes de una difusión desde (<code>irsrc,icsrc</code>) Recibe los datos provenientes de una difusión de una matriz triangular desde (<code>irsrc,icsrc</code>)
OPERACIONES COMBINADAS <code>a=gsum2d(a,rdest,cdest,[ictxt,scope,top,lda])</code> <code>a,ra,ca=gamx2d(a,rdest,cdest[,ictxt,scope,top,lda,rcflag])</code> <code>a,ra,ca=gamn2d(a,rdest,cdest[,ictxt,scope,top,lda,rcflag])</code>	 Devuelve la suma de los elementos de la matriz Devuelve el valor máximo del elemento y su posición en los datos Devuelve el valor mínimo del elemento y su posición en los datos

PyPBLAS Quick Reference

NIVEL 1

$x \longleftrightarrow y$
 $\alpha x \rightarrow x$
 $x \rightarrow y$
 $\alpha x + y \rightarrow y$
 $x^T y \rightarrow y$
 $x^T y \rightarrow y$
 $x^H y \rightarrow y$
 $\|x\|_2 \rightarrow y$
 $\|re(x)\|_2 + \|img(x)\|_2 \rightarrow asum$
 $1^{st} k \ni |re(x_k)| + |img(x_k)| \rightarrow indx$
 $max|re(x_i)| + |img(x_i)| \rightarrow amax$

```

x,y= pvswap(x,y)
x= pvscal(alpha,x)
y= pvcopy(x)
y= pvaxpy(alpha,x,y)
dot= pvdot(x,y)
dot= pvdotu(x,y)
dot= pvdotc(x,y)
nrm2= pvnrm2(x)
asum=pvasum(x)
amax,indx= pvamax(x)

```

NIVEL 2

$\alpha \cdot op(A) \cdot x + \beta \cdot y \rightarrow y$
 $\alpha \cdot A \cdot x + \beta \cdot y \rightarrow y$
 $\alpha \cdot A \cdot x + \beta \cdot y \rightarrow y$
 $A \cdot x \rightarrow x; A^T \cdot x \rightarrow x; A^H \cdot x \rightarrow x$
 $A^{-1} \cdot x \rightarrow x; A^{-T} \cdot x \rightarrow x; A^{-H} \cdot x \rightarrow x$
 $\alpha \cdot x \times y^T + A \rightarrow A$
 $\alpha \cdot x \times y^T + A \rightarrow A$
 $\alpha \cdot x \times y^H + A \rightarrow A$
 $\alpha \cdot x \times x^H + A \rightarrow A$
 $\alpha \cdot x \times y^H + y(alpha \cdot x)^H + A \rightarrow A$
 $\alpha \cdot x \times x^T + A \rightarrow A$
 $\alpha \cdot x \times y^T + y(\alpha \cdot x)^T + A \rightarrow A$

```

y=pvgemv(alpha,a,x,beta,y[,trans])
y=pvhemv(alpha,a,x,beta,y[,uplo])
y=pvsymv(alpha,a,x,beta,y)
x=pvtrmv(a,x,[uplo,trans,diag])
x= pvtrsv(a,x,[uplo,trans,diag])
a=pvger(alpha,x,y,a)
a=pvgeru(alpha,x,y,a)
a=pvgerc(alpha,x,y,a)
a=pvher(alpha,x,a,[uplo])
a=pvher2(alpha,x,y,a,[uplo])
a=pvsyr(alpha,x,a,[uplo='U'])
a=pvsyr2(alpha,x,y,a,[uplo])

```

NIVEL 3

$\alpha \cdot op(A) \times op(B) + \beta \cdot C \rightarrow C$
 $\alpha \cdot A \times B + \beta \cdot C \rightarrow C; \alpha \cdot B \times A + \beta \cdot C \rightarrow C$
 $\alpha \cdot A \times B + \beta \cdot C \rightarrow C; \alpha \cdot B \times A + \beta \cdot C \rightarrow C$
 $\alpha \cdot A \times A^T + \beta \cdot C \rightarrow C; \alpha \cdot A^T \times A + \beta \cdot C \rightarrow C$
 $\alpha \cdot A \times A^H + \beta \cdot C \rightarrow C; \alpha \cdot A^H \times A + \beta \cdot C \rightarrow C$
 $\alpha \cdot A \times B^T + \alpha \cdot B \times A^T + beta \cdot C \rightarrow C$
 $\alpha \cdot A \times B^H + \alpha \cdot B \times A^H + beta \cdot C \rightarrow C$
 $\beta \cdot C + \alpha \cdot A^T + \rightarrow C$
 $\beta \cdot C + \alpha \cdot A^T + \rightarrow C$
 $\beta \cdot C + \alpha \cdot A^H + \rightarrow C$
 $\alpha \cdot op(A) \times B \rightarrow B; \alpha \cdot B \times op(A) \rightarrow B$
 $\alpha \cdot op(A^{-1}) \times B \rightarrow B; \alpha B \cdot B \times op(A^{-1}) \rightarrow B$

```

c= pvgemm(alpha,a,b,beta,c,[transa,transb])
c=pvsymm(alpha,a,b,beta,c[,side,uplo])
c=pvhemm(alpha,a,b,beta,c[,side,uplo])
c=pvsyrk(alpha,a,beta,c[,uplo,trans])
c=pvherk(alpha,a,beta,c[,uplo,trans])
c=pvsyr2k(alpha,a,b,beta,c[,uplo,trans])
c=pvher2k(alpha,a,b,beta,c[,uplo,trans])
c=pvtran(alpha,a,beta,c)
c=pvtranu(alpha,a,beta,c)
c=pvtranc(alpha,a,beta,c)
b=pvtrmm(alpha,a,b[,side,uplo,transa,diag?])
b=pvtrsm(alpha,a,b[,side,uplo,transa,diag])

```

PyScaLAPACK Quick Reference

RUTINAS SENCILLAS

Rutinas sencillas para ecuaciones lineales

Tipo de Matriz	Rutina
General	<code>b, info=pvgesv(a, b)</code>
General en Bandas (sin pivotación)	<code>b, info=pvdbsv(a, b[bwl=n-1, bwu=n-1])</code>
General en Bandas (pivotación parcial)	<code>b, info=pvgbsv(a, b[bwl=n-1, bwu=n-1])</code>
General Tridiagonal	<code>b, info=pvdtsv(a, b)</code>
Simétrica/Hermitiana (Definida positiva)	<code>b, info=pvposv(a, b, [uplo='U'])</code>
Simétrica/Hermitiana (Definida positiva en bandas)	<code>b, info=pvpbsv(a, b, [uplo='U', bw=n-1])</code>
Simétrica/Hermitiana (Definida positiva Tridiagonal)	<code>b, info=pvptsv(d, e, b)</code>

Rutinas sencillas para problemas de mínimos cuadrados estándar

Tipo de Problema	Rutina
Factorización ortogonal asumiendo rango completo	<code>b, info= pvgels(a, b, [trans='N'])</code>

Rutinas sencillas para problemas de valores singulares y valores propios

Tipo de Problema/Matriz	Rutina
Simétrica/Hermitiana (Valores y vectores únicos)	<code>w, z, info= pvsyev(a[, jobz='N', uplo='U'])</code>
General (Valores y vectores singulares)	<code>s, u, vt, info= pvgesvd(a[, jobu='N', jobvt='N'])</code>

RUTINAS AVANZADAS

Rutinas avanzadas para ecuaciones lineales

Tipo de Matriz	Rutina
General	<code>b, info=pvgesvx(a, b[, fact='N', trans='N', equed='N', af=None, rcond=0])</code>
Simétrica/Hermitiana (Definida positiva)	<code>a, af, equed, sr, sc, b, x, rcond, ferr, berr, info=pvposvx(a, b[, fact='N', uplo='U', equed='N'])</code>

Rutinas sencillas para problemas de valores singulares y valores propios

Tipo de Problema/Matriz	Rutina
Simétrica (Valores y vectores únicos)	<code>m, nz, w, z, ifail, iclustr, gap, info= pvsyevx(a[, jobz='N', range='A', uplo='U', orfac=0, rcond=0, vl=None, vu=None, il=None, iu=None, abstol=0])</code>
General (Valores y vectores singulares)	<code>s, u, vt, info= pvgesvd(a[, jobu='N', jobvt='N'])</code>
Simétrica (Valores y vectores únicos)	<code>m, nz, w, z, ifail, iclustr, gap, info= pvsygvx(a[, jobz='N', range='A', uplo='U', orfac=0, rcond=0, vl=None, vu=None, il=None, iu=None, abstol=0])</code>
