

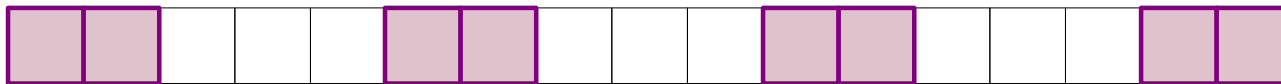
Tipus de Dades Derivats Regulars: Exemples

```
MPI_Type_vector(count, length, stride, type, newtype)
```

Crea un tipus de dades homogeni i regular a partir d'elements d'un *array* equiespaiats

- 1 De quants blocs es compona (*count*)
- 2 De quina longitud són els blocs (*length*)
- 3 Quina separació hi ha entre un element d'un bloc i el mateix element del següent bloc (*stride*)
- 4 De quin tipus són els elements individuals (*type*)

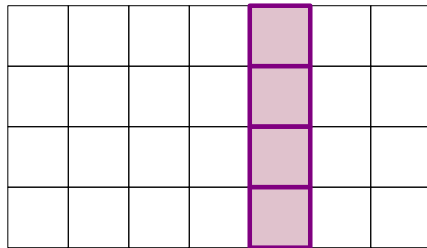
Exemple: tipus de dades necessari per a enviar/rebre els elements de color (d'un vector d'enters)



```
MPI_Datatype tipus;  
MPI_Type_vector(4,2,5,MPI_INT,&tipus);
```

Tipus de Dades Derivats Regulars: Exemples

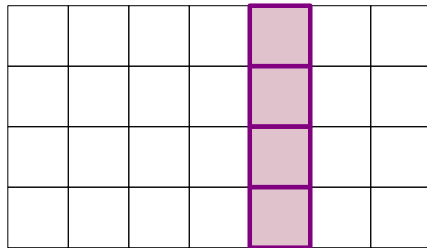
- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre la columna 4 en compte de la 2.



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

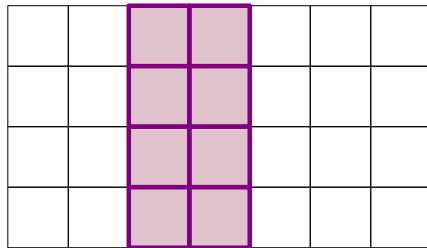
- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre la columna 4 en compte de la 2.



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envia la 3a columna */
    MPI_Send(&A[0][4], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][4], 1, columna, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

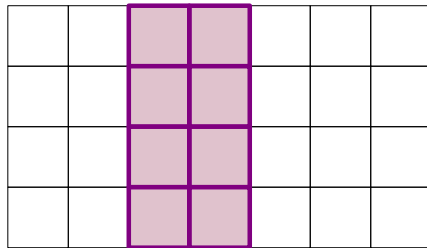
- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre el bloc de les columnes 2 i 3.



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                                /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

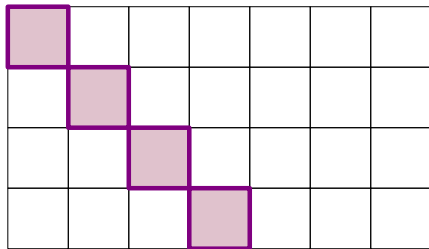
- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre el bloc de les columnes 2 i 3.



```
double A[4][7];
MPI_Datatype bloc;
MPI_Type_vector(4, 2, 7, MPI_DOUBLE, &bloc);
MPI_Type_commit(&bloc);
if (my_rank == 0) { /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, bloc, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, bloc, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

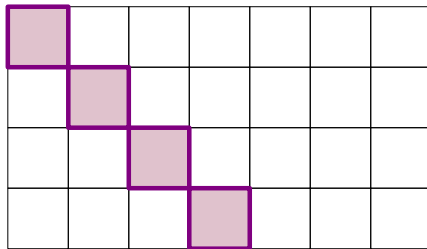
- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre els elements de la diagonal.



```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) { /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

- Sobre l'exemple de la transparència 46, modificar el codi per a enviar/rebre els elements de la diagonal.



```
double A[4][7];
MPI_Datatype diag;
MPI_Type_vector(4, 1, 8, MPI_DOUBLE, &diag);
MPI_Type_commit(&diag);
if (my_rank == 0) { /* envia la 3a columna */
    MPI_Send(&A[0][0], 1, diag, 1, 0, comm);
} else {
    MPI_Recv(&A[0][0], 1, diag, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

- Sobre l'exemple de la transparència 46, modificar el codi per a enviar la columna 2, però rebent els elements de forma contigua sobre un array `double b[4]`.

```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```


Tipus de Dades Derivats Regulars: Exemples

- Sobre l'exemple de la transparència 46, modificar el codi per a enviar la columna 2, però rebent els elements de forma contigua sobre un array `double b[4]`.

```
double A[4][7], b[4];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(b, 4, MPI_DOUBLE, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

- Sobre l'exemple de la transparència 46, modificar el codi per a enviar un array `double b[4]`, però rebent els elements sobre la columna 2 de la matriu A.

```
double A[4][7];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                /* envia la 3a columna */
    MPI_Send(&A[0][2], 1, columna, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```

Tipus de Dades Derivats Regulars: Exemples

- Sobre l'exemple de la transparència 46, modificar el codi per a enviar un array `double b[4]`, però rebent els elements sobre la columna 2 de la matriu A.

```
double A[4][7], b[4];
MPI_Datatype columna;
MPI_Type_vector(4, 1, 7, MPI_DOUBLE, &columna);
MPI_Type_commit(&columna);
if (my_rank == 0) {                /* envia la 3a columna */
    MPI_Send(b, 4, MPI_DOUBLE, 1, 0, comm);
} else {
    MPI_Recv(&A[0][2], 1, columna, 0, 0, comm, &status);
}
```