

## QCMATRIX APIs

```
typedef struct {  
    QInt dim_block;  
    RealMat **blocks;  
} QcMat;
```

```
QErrorCode QcMatWrite(QcMat* A,  
    const QChar *mat_label,  
    const QViewOption view_option)
```

### QCMATRIX Fortran Adapter (C part)

```
typedef struct {  
    QVoid *f03_mat;  
    QBool external_mat;  
} RealMat;
```

```
QErrorCode RealMatWrite(RealMat *A,  
    const QChar *mat_label,  
    const QViewOption view_option)  
{  
    ... ...;  
    Mat_Ptr_Write(&A->f03_mat, ...);  
    ... ...;  
}
```

`c_f_pointer()`

### QCMATRIX Fortran Adapter (Fortran part)

```
type(C_PTR) :: A
```

```
subroutine Mat_Ptr_Write(A, ...) bind(C, name="Mat_Ptr_Write")  
    use iso_c_binding  
    type(C_PTR), intent(in) :: A  
    ... ..  
    type(LANG_F_MATRIX), pointer :: A_ext  
    ... ..  
    call c_f_pointer(A, A_ext)  
    call Matrix_Write(A_ext, ...)  
    ... ..  
end subroutine Mat_Ptr_Write
```