[3.27] Fast symbolic determinants

The TI-89/92+ may take quite a while to find some symbolic determinants with *det()*. For example, a HW2 TI-92+ takes about 11 seconds to return the determinant of this matrix:

$$\begin{bmatrix} a & b & c \\ d & \sin(e) & f \\ g & \sqrt{h} & i \end{bmatrix}$$

Timité Hassan has written an excellent replacement for *det()* called *det2()*, which returns the determinant in about a second. You can get this routine at http://www.ti-cas.org. You will see similar execution time improvements when the matrix elements are exact numeric expressions involving square roots. Some floating-point determinants may be calculated more accurately.

I made some minor changes to Timité's code; I call this new function *det2a()*. I changed the local variable names, which reduced the code size considerably. I also defined the function *is_0()* within the function, so it is self-contained. I changed the returned error message strings. My modified routine is shown below.

```
det2a(m)
Func
©(mat) symbolic determinant
©T. Hassan
©Modified 4nov01/dburkett@infinet.com
Local n,k,i,j,q,p,r,is_0

Define is_0(z)=Func                  © Define local function
 when(z=0,true,false,false)
EndFunc

If getType(m)≠"MAT"                  © Test input argument type
 Return  "det2a:Invalid argument"

dim(m)→n                             © Get matrix dimensions
If n[1]≠n[2]                         © Test for square matrix
 Return "det2a:Not square matrix"

n[1]→n                               © Get number of matrix rows
If n=1 Then                          © 1x1 matrix
Return m[1,1]
ElseIf n=2 Then                      © 2x2 matrix
 Return m[1,1]*m[2,2]-m[2,1]*m[1,2]
ElseIf n=3 Then                      © 3x3 matrix
 Return
m[1,1]*(m[2,2]*m[3,3]-m[2,3]*m[3,2])-m[1,2]*(m[2,1]*m[3,3]-m[2,3]*m[3,1])+m[1,3]*(m[2,1]*
m[3,2]-m[2,2]*m[3,1])
EndIf

1→q                                  © N x N matrix, N>3
For i,1,n-3
 m[i,i]→p
 If is_0(p) Then
  For j,i+1,n
   If not is_0(m[j,i]) Then
    rowSwap(m,i,j)→m
    Exit
   EndIf
  EndFor

  If j>n
   Return  0
 EndIf
```

```
 For j,i+1,n
  m[j,i]→r
  If not is_Ø(r) Then
   r*m[i]-p*m[j]→m[j]
   q*⁻p→q
  EndIf
 EndFor
EndFor

Return
1/q*Π(m[k,k],k,1,n-3)*(m[n-2,n-2]*(m[n-1,n-1]*m[n,n]-m[n-1,n]*m[n,n-1])-m[n-2,n-1]*(m[n-1
,n-2]*m[n,n]-m[n-1,n]*m[n,n-2])+m[n-2,n]*(m[n-1,n-2]*m[n,n-1]-m[n-1,n-1]*m[n,n-2]))

EndFunc
```