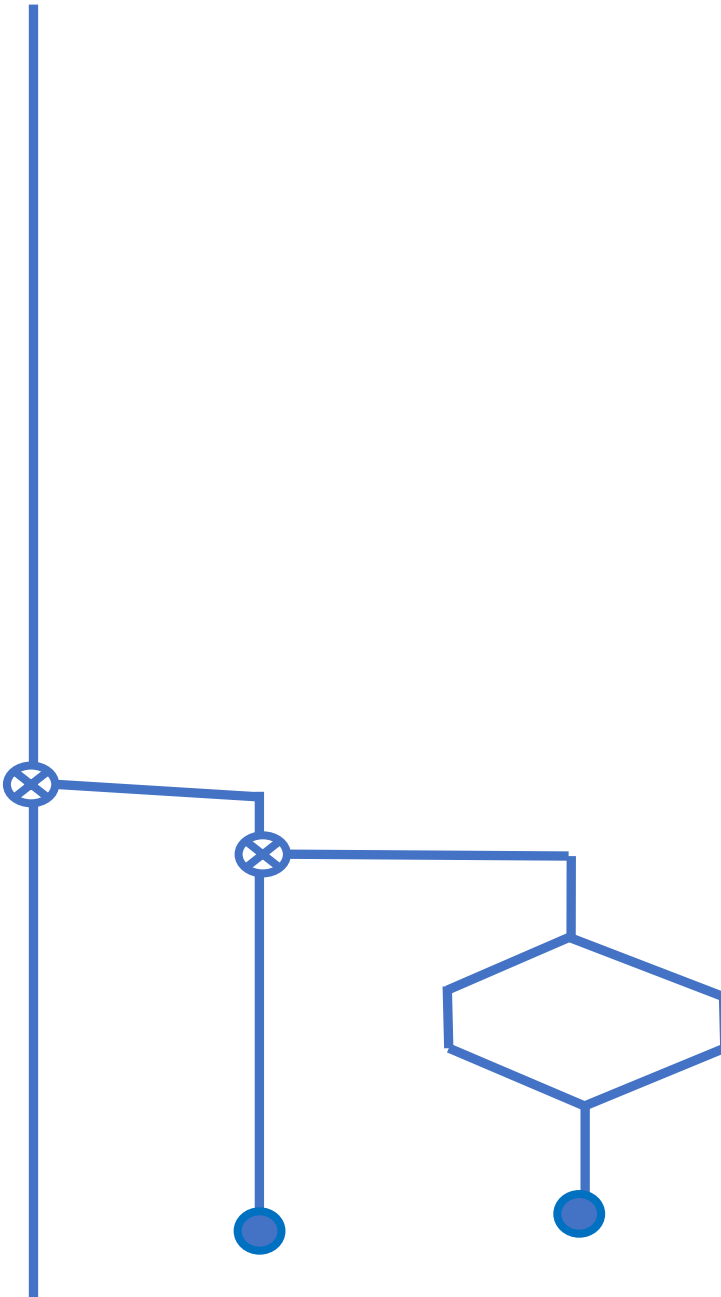


# Living In The Matrix

Forth Day

Nov. 21, 2020

Bill Ragsdale



# Today We'll Cover

- Traditional matrix syntax.
- A Forth appropriate syntax.
- A Forth matrix extension word set.
- Operation on matrices.
- Matrix calculations.

# Credits

- Andrew McKewan and Tom Zimmer for Win32Forth.
- The European team who updated it in the early 2000s.
- Dr. Julian Noble for the key concepts of matrix structure and its parameter use. See *Scientific FORTH*

# Creating Float Matrix

```
3 4 create{F first{  
    first{ }list
```

```
.000000 .000000 .000000 .000000  
.000000 .000000 .000000 .000000  
.000000 .000000 .000000 .000000 ok
```

# Direct Data Entry

```
first{ { [ 1 1 1 1 | 2 2 2 2 | 3 3 3 3 ] }
```

```
first{ }list
```

```
1.0000 1.0000 1.0000 1.0000
```

```
2.0000 2.0000 2.0000 2.0000
```

```
3.0000 3.0000 3.0000 3.0000 ok
```

# Filling Values

```
first{ }fill
```

```
first{ }list
```

```
.00000 1.00000 2.00000 3.00000
```

```
4.00000 5.00000 6.00000 7.00000
```

```
8.00000 9.00000 10.0000 11.0000 ok
```

# Matrix Math

```
first{ first{ first{ }+  
                first{ }list  
that is: A  B  +  C  !
```

```
  .0000  2.0000  4.0000  6.0000  
 8.0000 10.0000 12.0000 14.0000  
16.0000 18.0000 20.0000 22.0000
```

## Transposing A Matrix

```
first{ }fill      first{ }list
.0000  1.0000  2.0000  3.0000
4.0000  5.0000  6.0000  7.0000
8.0000  9.0000 10.0000 11.0000
```

```
first{ }transpose  first{ }list
.00000  4.0000  8.0000
1.0000  5.0000  9.0000
2.0000  6.0000 10.0000
3.0000  7.0000 11.0000
```



# Traditional Matrix Syntax

```
C[1;3] = A[2;2] + B[2;3]
```

**A variable name**

**Cells within the matrix**

**Data operations on the values**

# What We'll Cover

## Traditional Matrix Semantics

$$C[1;3] = A[2;2] + B[2;3]$$

## In Forth:

A{ {{ 2 2 }} F@

B{ {{ 2 3 }} F@ F+

C{ {{ 1 3 }} F!

# The Good and the Bad

Traditional syntax is terse. But . . . .

Not clear in cases such as:

$C[ : ; 3:end] = A[ : ; 1\ 2\ 3](3 : 2) + B[1\ 2 ; 3\ 4]$

Forth factors out the stages and has a clear reverse Polish ordering.

# Picking It Apart

```
A{ {{ 2 3 }} F@
```

**In Forth:**

**A{** gets the address of a matrix

**{{** gets descriptor of that matrix  
(address rows columns b/cell)

**2 3** specify the row and column

**}}** gets the storage address

**F@** on operation on that address

# Picking It Apart

Stack values along the way:

```
A{      {{ 2 3 }}      F@  
  addr      r  c      addr  F: 1.200
```

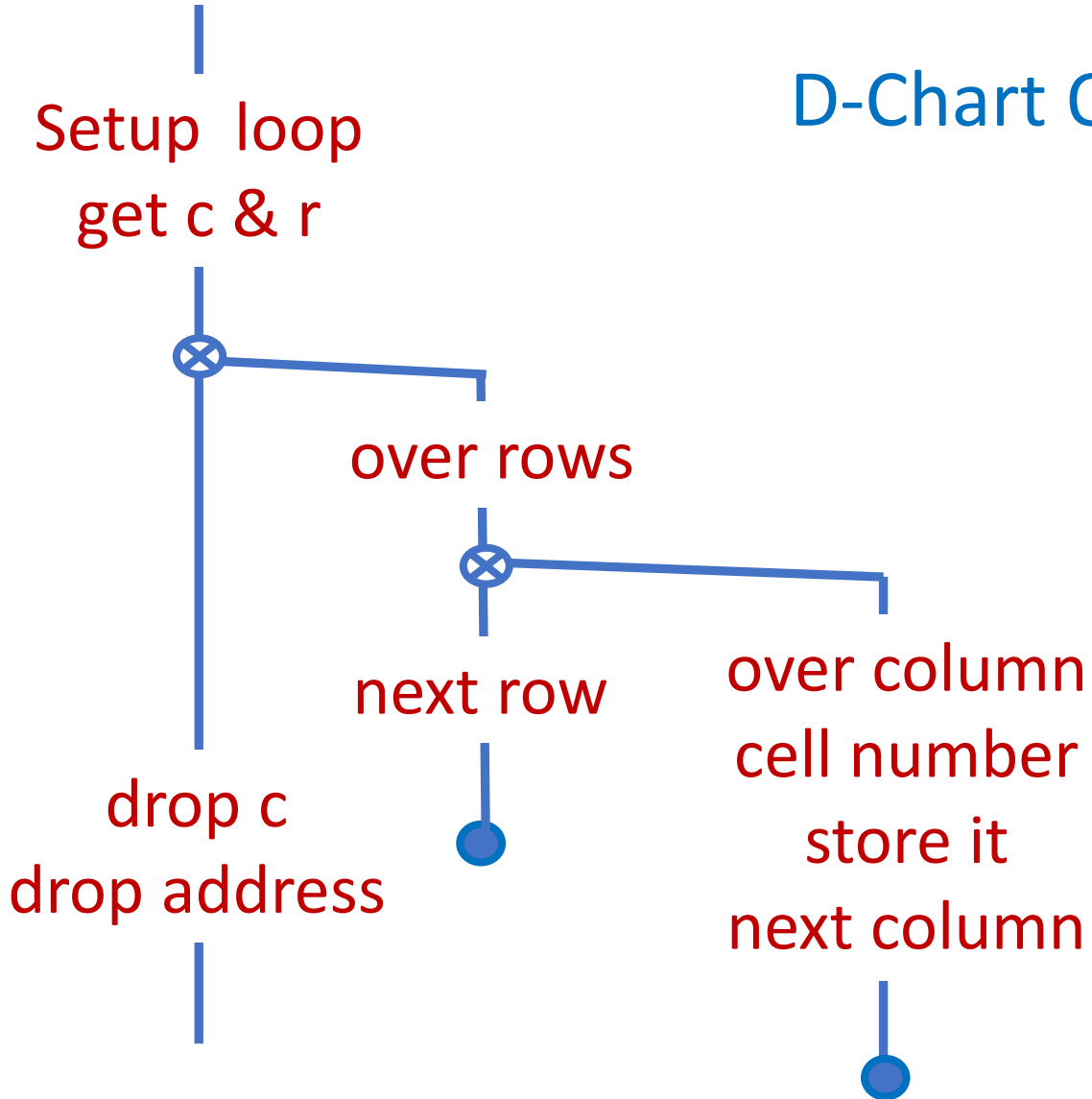
(And some housekeeping stack values  
you don't see.)

# Programming Example

```
: }Fill ( matrix_address --- )  
  dup }SetupLoop ( c r )  
  0 DO  
    dup 0 DO  
      dup j * i + s>f <<stored value  
      over{{ j i }}F! <<storing  
    LOOP  
  LOOP  
  2drop ;
```

Note: There is more on the stack you don't see.

# D-Chart Of }fill



# Some Details

Forth Matrix is zero based for rows & columns.

Words with a single { or } operate on a matrix address: first{ }list

The word {{ converts a matrix address into its 'descriptor': address rows columns b/cell

Words with a double {{ or }} operate a 'descriptor' plus row and column values.



# Word Summary

Create: create{ create{F

Support: }list }fill

Data Entry: {[ | ]}

Manipulation: }transpose }invert }det }eye

Matrix Math: }+ }- }\* }/ }dot\*

Row/Col: }}exchR }}exchC

Stack Access: dup{{ over{{ pick{{

Cell Math: {{ }} }}F@ }}F? }}F! }}F+!

# Levels

Level Zero

Internals

}rcConformal? }}Bytes

Level One

Cell to Cell

{{ }} }}F@

Level Two

By rows and/or columns

UNDER DEVELOPMENT

Level Three

Matrix level

create{F

}fill }list }copy

}+ }= }transpose }dot\*

# Summary

I use MatLab and the clone Octave.

I have a project about once a year and need a complete refresher on the semantics.

The manuals explain each single structure but not the overall parsing order.

Therefore I need a reverse Polish syntax that needs no 'refresher course'.

Matrix Forth is the answer.

# References

- <https://github.com/BillRagsdale/Matrix-Forth-Wordset>
- <https://github.com/BillRagsdale/WIN32Forth-Guide>