

# Numerical Scheming with `matrico`

Christian Himpe

Village CHICKENs 2023

# Who's that ?

- I have been a research mathematician for more than a decade.
- Now I am a data engineer by day ...
- ...and a `CHICKEN` Schemer by night 😊

# Why **CHICKEN** ?

- I somewhat lost hope in dev ... until I found **CHICKEN**
- My Scheme journey: **Racket** → **Chez** → **CHICKEN**
- My key features:  
**R5RS** , **chicken.base** , **SRFI-4** , typed math, modules, functors, profilers,  
cross platform, open-source, **excellent documentation**

# What's `matrico` ?

- `CHICKEN` egg
- Numerical environment, inspired by `MATLAB`, `Octave`, `Julia`, and `NumPy`
- Matrix library providing: matrix algebra, numerical linear algebra, numerical analysis
- Fully implemented in `CHICKEN` Scheme, particularly not wrapping `BLAS` or `LAPACK`

# Why `matrico` ?

- Learning Scheme
- Teaching numerics
- Contributing an egg
- "Scheme for Scientific Computing"
- `CHICKEN` is fun!

# What's a `matrico` Matrix ?

- A (dimension-immutable) capsuled record, of:
- a `list` of columns
- which are by default `f64vector`s (dense),
- and the number of columns.
- "Best of both worlds": lists & arrays

# Matrix algebra

- Constructors
- Arithmetic
- Accessors
- Predicates
- Mappers
- Folders

# Numerics

## Numerical Linear Algebra

- Matrix multiplication
- QR decomposition (Schwarz-Rutishauser algorithm)
  - Linear solver, Orthogonalization, Approximate determinant

## Numerical Analysis

- 2nd-order ODE solvers with adaptable stability region:
  - hyperbolic Runge-Kutta integrator
  - strong-stability-preserving Runge-Kutta integrator



# Some Tidbits

- Procedural constants (using [OEIS](#))
- One-based indexing (ie `MATLAB`) and negative indexing (ie `NumPy`)
- `matrico` function (online help)
- `mx` prefix (frontend namespace)
- Unicode matrix printing
- Native matrix `.scm` file format, `.csv` export

# How To `matrico` ?

```
; Solve heat equation (1D linear diffusion)
;  $\dot{x}(t) = Ax(t) =: f(t,x)$ 

(import matrico)

(define N 100)
(define h (/ 10.0 (add1 N)))
(define h^2 (* h h))

(define A^T (mx-tridiag N (/ 1.0 h^2) (/ -2.0 h^2) (/ 1.0 h^2)))

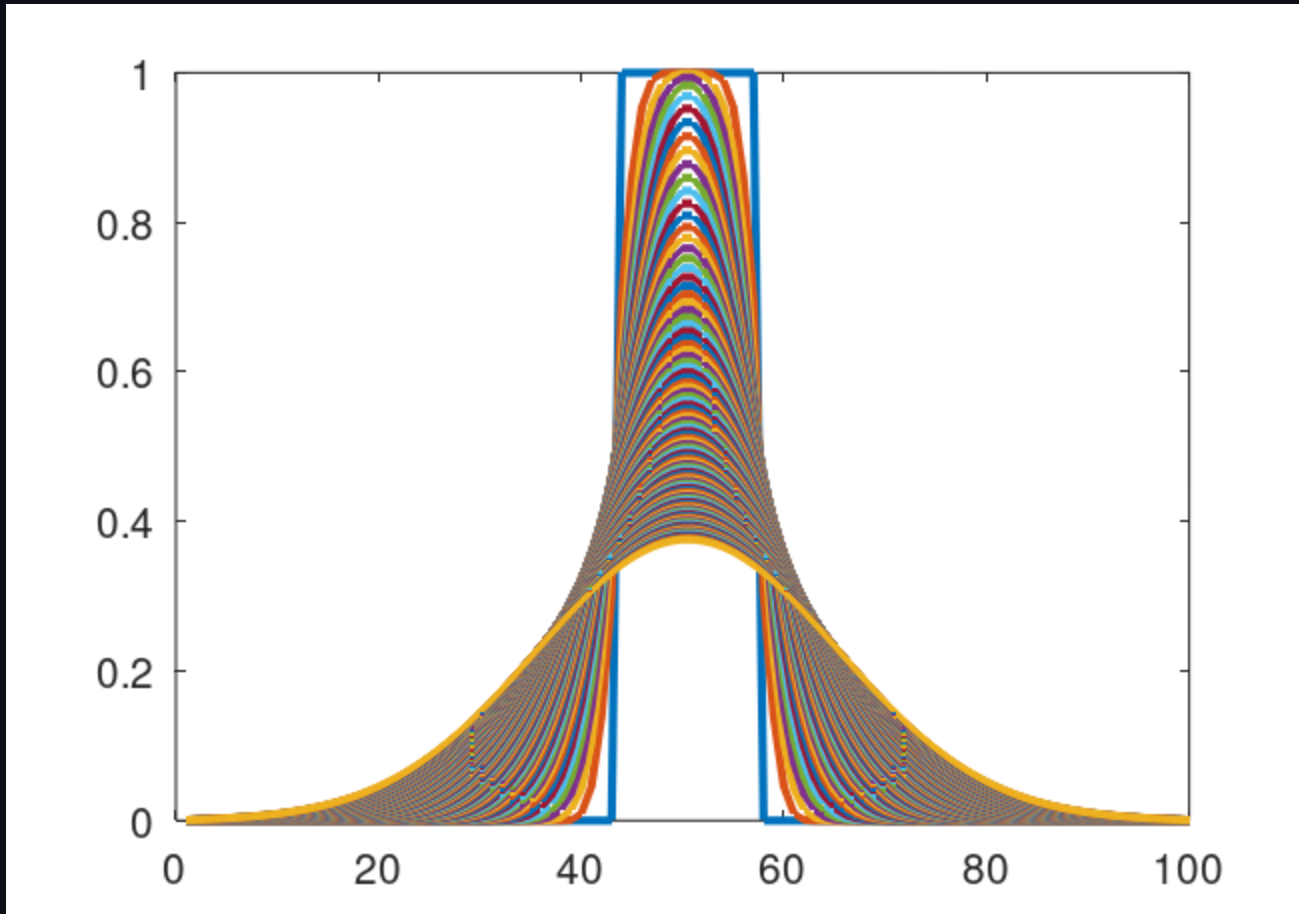
(define x0 (mx-vercat (mx 43 1 0.0) (mx-vercat (mx 14 1 1.0) (mx 43 1 0.0))))

(define (f t x)
  (mx-dot* A^T x))

(define X (mx-ode2-ssp 5 f (cons 0.01 1.0) x0))

(mx-export "heat.csv" X)
```

# Solution Over Time



```
octave> plot(load("heat.csv"), 'linewidth', 2)
```

# What's Next ?

- Rank revealing QR decomposition
- Pseudo inverse via QR decomposition
- Eigenvalue Decomposition (EVD) via QR Algorithm (?)
- Singular Value Decomposition (SVD) via QR (?)
- Wrap or reimplement [Unicode Plot](#)
- [Flame graphs](#) for profiler output (?)

# My **CHICKEN** Projects

- **matrico** (math) egg: <http://wiki.call-cc.org/eggref/5/matrico>
- **arcadedb** (database) egg: <http://wiki.call-cc.org/eggref/5/arcadedb>
- **CHICKEN Dockerfile** : [gist](#)

# That's All Folks

- Blog: <http://numerical-schemer.xyz>
- Website: <https://himpe.science>
- Github: [gramian](#)
- Dev.to: [gramian](#)
- Twitter: [@modelreduction](#)

Slides: <https://himpe.science/talks/himpe23-chicken.pdf>