Numerical computations for the tear film equations in a blink cycle with spectral collocation methods

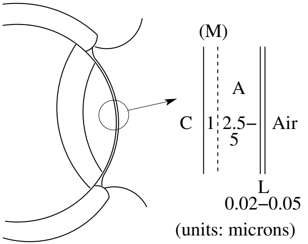
A. Heryudono[1], R.J. Braun[1], T.A. Driscoll[1], K.L. Maki [1], L.P. Cook[1], and P.E. King-Smith[2]

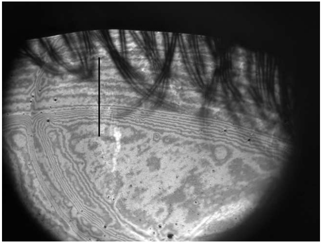
[1]Mathematical Sciences, U of Delaware
and [2]College of Optometry, Ohio State U

Applied Mathematics Seminar
George Mason University
October 26 2007

Outline

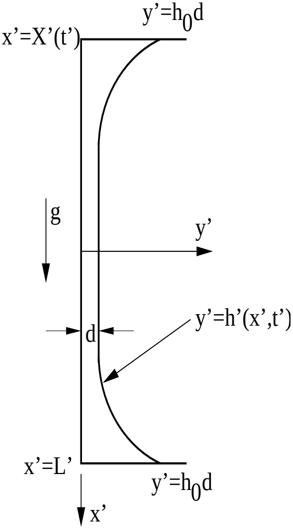
How do we simulate the dynamics of the tear film ?





Interference fringes.

Get insight from 1-D case first.



Physical parameters: Braun et al.

Constants	Description
$L' = 5$ mm	half the width of the palpebral fissure (x direction)
$d = 5$ μm	thickness of the tear film away from ends
$\epsilon = \frac{d}{L'} \approx 10^{-3}$	small parameter for lubrication theory
$U_m = 10$–30 cm/s	maximum speed across the film
$L'/U_m = 0.05$ s	time scale for real blink speeds
$\sigma_0 = 45$ mN/m	surface tension
$\mu = 10^{-3}$ Pa·s	viscosity
$\rho = 10^3$ kg/m^3	density

- Inside the film
  - Viscous incompressible parallel flow inside the film.
  - Inertial terms and gravity are neglected.

- Inside the film
  - Viscous incompressible parallel flow inside the film.
  - Inertial terms and gravity are neglected.
- At the wall
  - On the impermeable wall at $y = 0$, we have the boundary conditions

$$v = 0, \quad u = \beta u_y;$$

- Inside the film
  - Viscous incompressible parallel flow inside the film.
  - Inertial terms and gravity are neglected.
- At the wall
  - On the impermeable wall at $y = 0$, we have the boundary conditions

$$v = 0, \quad u = \beta u_y;$$

- At the free surface
  - Simplified normal stress condition at $y = h(x, t)$

$$p = -Sh_{xx}, \quad S = \frac{\epsilon^3 \sigma}{\mu U_m}$$

  - Kinematic condition

$$h_t + q_x = 0 \text{ on } X(t) \leq x \leq 1,$$

where

$$q = \int_0^h u(x, y, t)dy$$

- The stress-free case.

$$q(x,t) = Sh_{xxx}\left(\frac{h^3}{3} + \beta h^2\right)$$

Boundary conditions

$$h(X(t), t) = h(1, t) = h_0 \quad q(X(t), t) = X_t h_0 + Q_{top} \quad q(1, t) = -Q_{bot}.$$

$$h_t + q_x = 0 \text{ on } X(t) \le x \le 1,$$

where

$$q = \int_0^h u(x,y,t)dy$$

- The stress-free case.

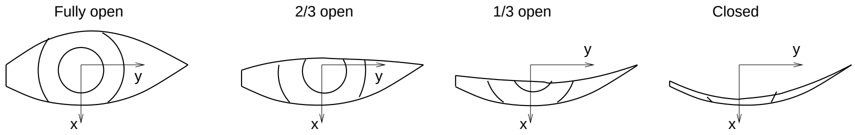
$$q(x,t) = Sh_{xxx}\left(\frac{h^3}{3} + \beta h^2\right)$$

- The uniform stretching limit (USL).

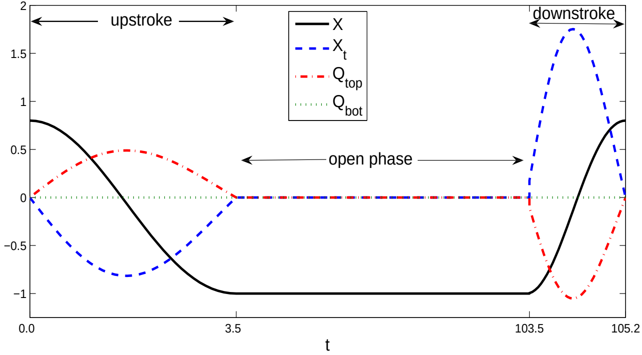
$$q(x,t) = \frac{h^3}{12}\left(1 + \frac{3\beta}{h+\beta}\right)(Sh_{xxx}) + X_t \frac{1-x}{1-X}\frac{h}{2}\left(1 + \frac{\beta}{h+\beta}\right)$$

Boundary conditions

$$h(X(t),t) = h(1,t) = h_0 \quad q(X(t),t) = X_t h_0 + Q_{top} \quad q(1,t) = -Q_{bot}.$$



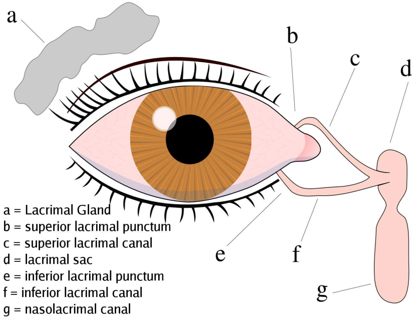
Berke and Mueller (98), Heryudono et al (07)



- Flux proportional to lid motion (FPLM) (Jones et al (05))

$$Q_{top} = -X_t h_e, \quad Q_{bot} = 0$$

- Add in lacrimal gland supply and punctal drainage approximated by Gaussians.



Picture is taken from the Wikipedia commons

1. Moving boundary problem.
   - Fixed grid scheme is not convenient.

1. Moving boundary problem.
   - Fixed grid scheme is not convenient.
2. Fourth-order in space.
   - Roundoff errors in computing high derivatives.
   - Stiffness.

1. Moving boundary problem.
   - Fixed grid scheme is not convenient.
2. Fourth-order in space.
   - Roundoff errors in computing high derivatives.
   - Stiffness.
3. Nonlinearity.
   - Implicit method.

1. Moving boundary problem.
   - Fixed grid scheme is not convenient.
2. Fourth-order in space.
   - Roundoff errors in computing high derivatives.
   - Stiffness.
3. Nonlinearity.
   - Implicit method.
4. Third-order boundary conditions.
   - Imposing boundary conditions (BCs).
   - Stability.

1. Moving boundary problem.
   - Fixed grid scheme is not convenient.
2. Fourth-order in space.
   - Roundoff errors in computing high derivatives.
   - Stiffness.
3. Nonlinearity.
   - Implicit method.
4. Third-order boundary conditions.
   - Imposing boundary conditions (BCs).
   - Stability.
5. Variable resolution and/or accurate high-order derivatives near boundaries.
   - Adaptive scheme may be needed.

We transform the PDE into a fixed domain $[-1, 1]$ via

$$\xi = 1 - 2\frac{1-x}{1-X(t)}.$$

The equations (e.g. Stress free case) become

$$H_t = \frac{1-\xi}{L-X} X_t H_\xi - \left(\frac{2}{L-X}\right) Q_\xi$$

$$Q = S\left(\frac{2}{L-X}\right)^3 \left(\frac{H^3}{3} + \beta H^2\right) H_{\xi\xi\xi}$$

$$H(\pm 1, t) = h_0,\ \ Q(-1, t) = X_t h_0 + Q_{top},\ \ Q(1, t) = -Q_{bot},$$

$$H(\xi, 0) = h_m + (h_0 - h_m)\xi^m.$$

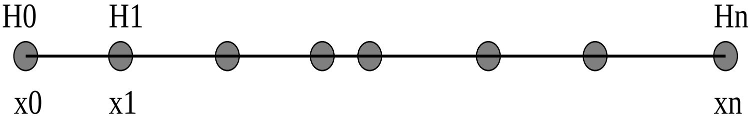
$\xi \in [-1, 1]$.

Advantages

- Global high accuracy for smooth function.
- Fast matrix-vector algorithm via FFT.
- Powerful theory (potential theory, orthogonal functions)

Disadvantages

- Dense differentiation matrices.
- Must use nodes with special distributions.
- Hard to apply in problems involving irregular geometry.

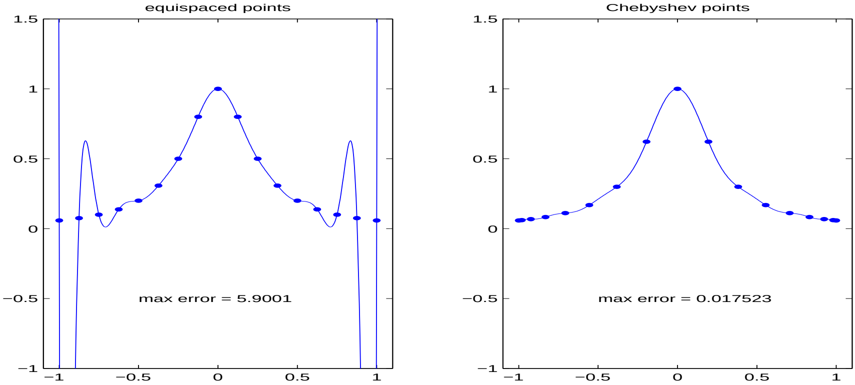


$$H(x) = \sum_{j=0}^{n} H_j \ell_j(x), \quad \ell_j = \frac{\prod_{k=0,k\neq j}^{n}(x - x_k)}{\prod_{k=0,k\neq j}^{n}(x_j - x_k)}$$

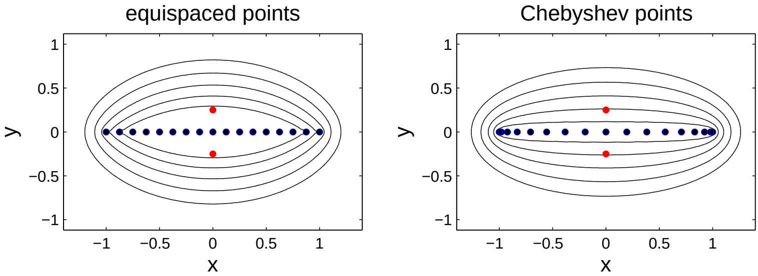
The Lagrange polynomial ℓ_j corresponding to the node x_j has the property

$$\ell_j(x_k) = \begin{cases} 1 & , \quad j = k, \\ 0 & , \quad \text{otherwise}, \end{cases} \qquad j, k = 0..., n.$$

Node locations $x_j, \quad j = 0, 1, \ldots, N$	Comments
$x_j = -1 + \frac{2j}{N}$	equi-spaced
$x_j = -\cos\left(\frac{\pi j}{N}\right)$	Chebyshev



Interpolating $f(x) = 1/(1 + 16x^2)$ with $N = 16$ nodes



Analyticity of $f(x)$ inside the region bounded by the smallest equipotential curve that contains $[-1, 1]$

$$H'(x) = \sum_{j=0}^{n} H_j \ell_j'(x), \quad H''(x) = \sum_{j=0}^{n} H_j \ell_j''(x)$$

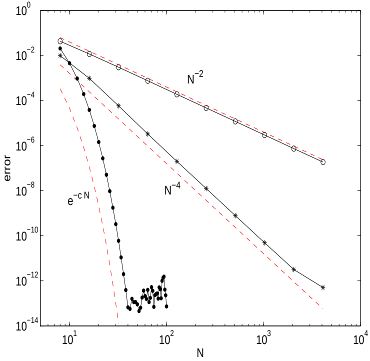
Computing k-th derivative $\Rightarrow$ Matrix-Vector product

$$\begin{bmatrix} & & \\ & D^{(k)} & \\ & & \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} u_0^{(k)} \\ \vdots \\ u_N^{(k)} \end{bmatrix},$$

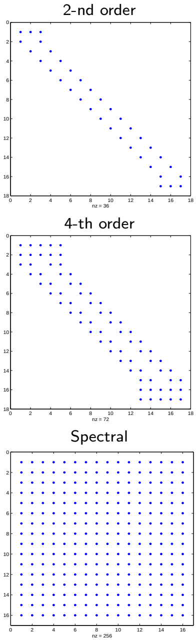
where,

$$D_{ij}^{(1)} = \ell_j'(x_i), \quad D_{ij}^{(2)} = \ell_j''(x_i), \quad \text{etc.}$$

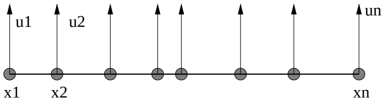
Trefethen (2000), Spectral Methods in MATLAB, Welfert (97), Baltensperger & Trummer (02)



Spectral convergence for the first derivative of $f(x) = 1/(1 + 16x^2)$



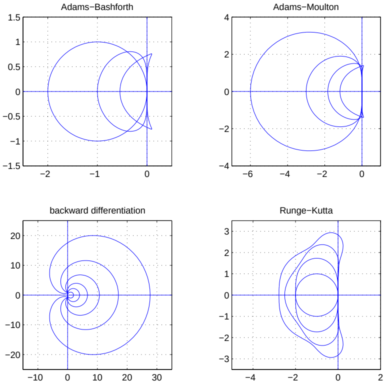
Spectral discretization in space and standard ODE in time.



example :

$$u_t = u_x \quad x \in [-1, 1)$$
$$u(1, t) = 0$$
$$\Downarrow$$
$$\hat{u}_t = \tilde{D}\hat{u}$$

$\Delta t \Lambda(\tilde{D}) \Rightarrow$ stability



Properties.

- Extreme eigenvalues of Chebyshev differentiation matrices: $O(N^2)$.
- Minimal spacing of N Chebyshev nodes:
  $\Delta x_{min} = 1 - \cos(\pi/N) = O(N^{-2})$.
- Explicit time marching scheme $\rightarrow \Delta t = O(N^{-2})$.

Kosloff & Tal-Ezer (93)

- Map Chebyshev points to a set points with larger minimal spacing.

Properties.

- Extreme eigenvalues of Chebyshev differentiation matrices: $O(N^2)$.
- Minimal spacing of N Chebyshev nodes: $\Delta x_{min} = 1 - \cos(\pi/N) = O(N^{-2})$.
- Explicit time marching scheme $\rightarrow \Delta t = O(N^{-2})$.

Kosloff & Tal-Ezer (93)

- Map Chebyshev points to a set points with larger minimal spacing.
- Roundoff reduction

Consider a symmetric transformation

$$\psi = g(\xi;\alpha) = \frac{\sin^{-1}(\alpha\xi)}{\sin^{-1}(\alpha)}, \quad \psi,\xi \in [-1,1], \quad \alpha \in (0,1).$$

By using chain rule, we obtain

$$\frac{df}{d\psi} = \frac{1}{g'(\xi;\alpha)}\frac{df}{d\xi}$$

for any given $f \in C^1[-1,1]$.

$$\alpha = 1 - \frac{c}{N^2} + O(N^{-3}), \quad c > 0 \;\Rightarrow \Delta\psi_{min} = O(N^{-1})$$

$$\psi = g(\xi;\alpha,\beta) = \frac{1}{a}\left(\sin^{-1}\left(\frac{2\alpha\beta\xi+\alpha-\beta}{\alpha+\beta}\right) - b\right), \quad \psi,\xi \in [-1,1]$$

where α and β control distribution points near $\xi = 1$ and $\xi = -1$ respectively. If $\alpha = \beta$, we end up having back to standard symmetric mapping.

Properties.

- Roundoff error in computing derivatives with Chebyshev differentiation matrices:

$$O(N^{2k}),$$

where k is the order of derivative.

- Mostly happens near boundaries.

Don & Solomonoff (97)

Choice of parameter :

$$\alpha = \operatorname{sech}\left(\frac{|ln\epsilon|}{N}\right),$$

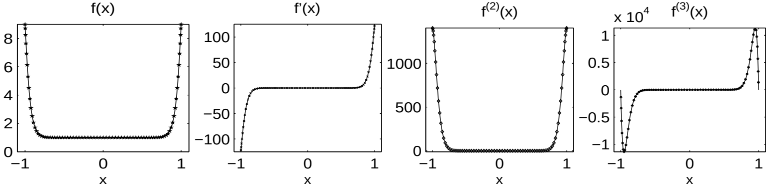
where ϵ is the machine precision.

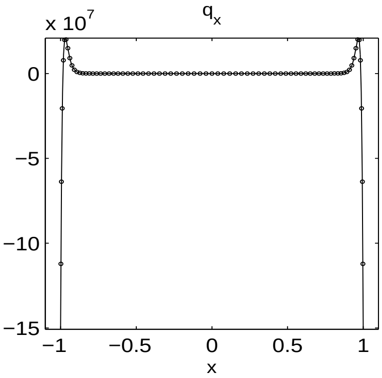
Roundoff error reduction $\Rightarrow$ $O((N\,|\ln\epsilon|)^k)$

Let $f(x)$ be a smooth function on the interval $[-1, 1]$ defined by

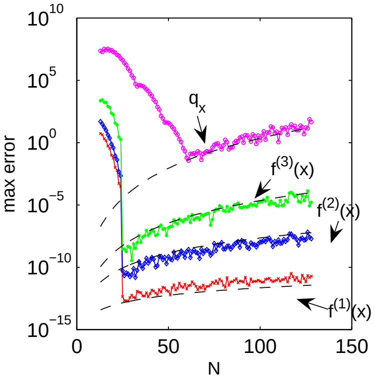
$$f(x) = 1 - \frac{h_0 - 1}{6(m+1)}(2m^2(x^2 - 1) + m(x^2 - 7) - 6)x^{2m+2}.$$

The graphs of $f(x)$ for $m = 10$ and its derivatives.



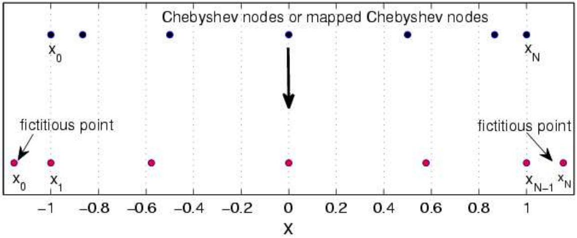


$$q_x = ((\frac{f^3}{3} + \beta f^2) f^{(3)})_x$$



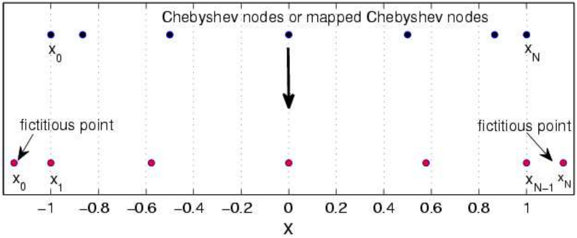
Dashed lines are the theoretical roundoff effects

Fornberg (06)



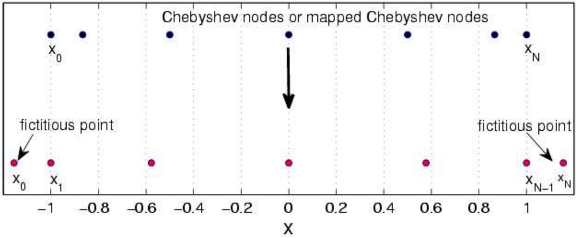
- Scale Chebyshev points such that x_1 and x_{N-1} become -1 and 1 respectively.

Fornberg (06)



- Scale Chebyshev points such that x_1 and x_{N-1} become -1 and 1 respectively.
- Form interpolant based on new nodes.

Fornberg (06)



- Scale Chebyshev points such that x_1 and x_{N-1} become -1 and 1 respectively.
- Form interpolant based on new nodes.
- Solve u_0 and u_N in terms of unknown u_j based on boundary conditions. Then incorporate them in the differentiation matrices.

H at discretized points.

$$H = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{N-1} \\ H_N \end{bmatrix}$$

k-th partial derivatives of $H(\xi, t)$ at unknown points $\xi_2..\xi_{N-2}$ can be written as

$$H^{(k)} = D^{(k)} H$$

$D^{(k)}$ already contains information about H_0, H_1, H_{N-1} and H_N which are obtained from boundary conditions.

$\Rightarrow$ In our simulations, only works for linear case.

H and Q each has its own interpolant. No fictitious points needed.

$$H = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_{N-1} \\ H_N \end{bmatrix} \qquad Q = \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{N-1} \\ Q_N \end{bmatrix}.$$

$$H^{(k)} = D^{(k)} H \quad \text{and} \quad Q^{(k)} = D^{(k)} Q$$

H_0, H_N, Q_0 and Q_N are known from boundary conditions and hence only values of H at inner nodes ξ_i, where $i = 1, .., N-1$, need to be found.

$\Rightarrow$ The system of eqs is twice as large

$$M\begin{bmatrix}H_t\\Q_t\end{bmatrix}=\begin{bmatrix}\mathrm{I} & 0\\0 & 0\end{bmatrix}\begin{bmatrix}H_t\\Q_t\end{bmatrix}=\begin{bmatrix}AD^{(1)}H+BD^{(1)}Q\\C(\frac{1}{3}H^3+\beta H^2)(D^{(3)}H)-Q\end{bmatrix}$$

A, B, and C are all $(N-1)\times(N-1)$ matrix with elements

$$A_{ij}=\frac{1-\xi_i}{L-X(t)},\quad B_{ij}=\frac{-2}{L-X(t)},\quad C_{ij}=S\left(\frac{2}{L-X(t)}\right)^3.$$

for all $i,j=1,..,N-1$. M is a singular $2(N-1)\times 2(N-1)$ matrix called mass matrix.

$\Rightarrow$ Solve DAE of index 1 with DASPK or ode15s in MATLAB.

- No fictitious points needed.
- Set Q

$$Q = X_t \frac{1-\xi}{2} \frac{H}{2} \left(1 + \frac{\beta}{H+\beta}\right) + \frac{H^3}{12} \left(1 + \frac{3\beta}{H+\beta}\right) \left[S \left(\frac{2}{1-X}\right)^3 H_{\xi\xi\xi} \right]$$

- When computing Q_ξ, use $Q(-1,t) = X_t h_0 + Q_{top}$, $Q(1,t) = -Q_{bot}$.
- Solve the initial value problem at inner nodes

$$H_t = \frac{1-\xi}{1-X} X_t H_\xi - \left(\frac{2}{1-X}\right) Q_\xi$$

with ode solver ode15s.

$\Rightarrow$ possible connection with penalty method.

Homogenization can be done by shifting variables H and Q such that

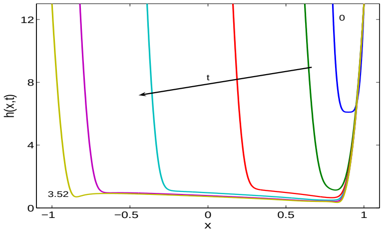
$$\hat{H} = H - h_0$$
$$\hat{Q} = Q - (a\xi + b).$$

It is clear that $\hat{H}(\pm 1, t) = 0$. In order to find a and b such that $\hat{Q}(\pm 1, t) = 0$, we end up solving 2×2 system of linear equations consisting of $a + b = Q(1, t)$ and $-a + b = Q(-1, t)$.

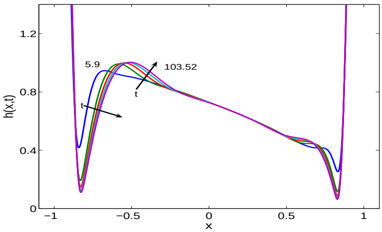
$\Rightarrow$ Our simulations work with/without homogenization.

Parameters $N = 351, \lambda = 0.1, \beta = 10^{-2}, S = 2 \times 10^{-5}, h_0 = 13, h_e = 0.6$, and initial volume $V_0 = 2.576$. Our simulation is done in MATLAB with ode15s as ODE solver.

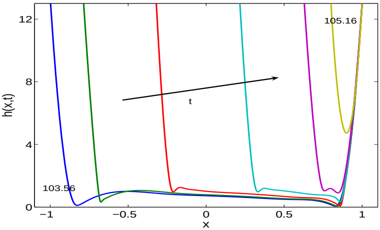
Opening phase

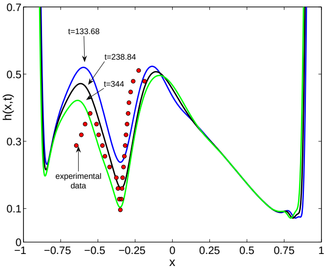


fully open (zoom)

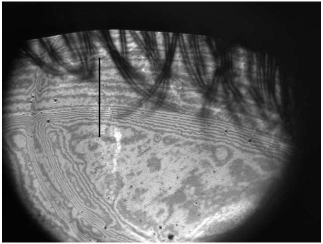


The closing phase

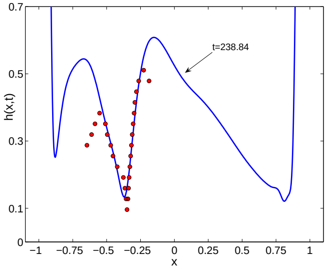




$S = 2 \times 10^{-5}$ case at various times

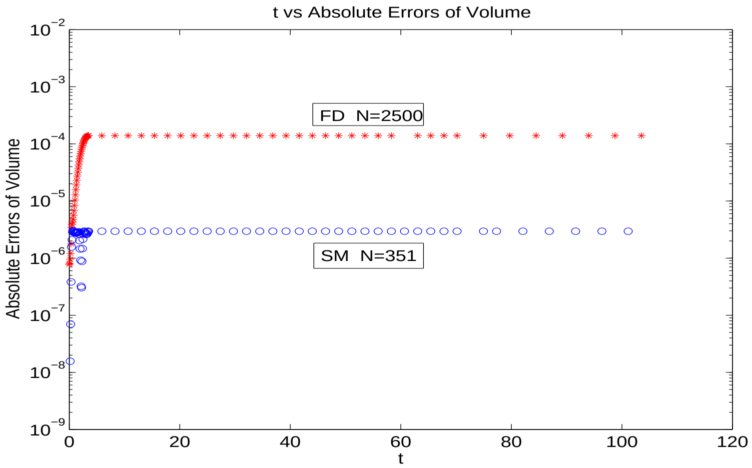


Interference fringes.



$S = 8 \times 10^{-6}$

More results $\Rightarrow$ Braun's talk today !!


t vs Absolute Errors of Volume
Absolute Errors of Volume
t
FD N=2500
SM N=351
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
0
20
40
60
80
100
120

- What happen with fictitious point method in the nonlinear case ?
- Adaptive radial basis functions with MOL (Driscoll & Heryudono (06)).
- Adaptive space-time radial basis functions.
- 2-D simulation.