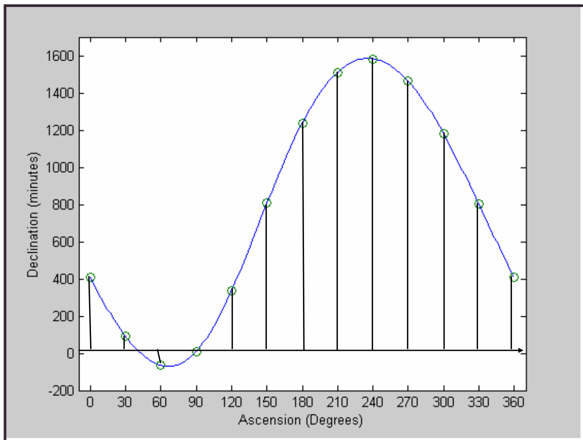
Math.375

IV- Interpolation 2 and beyond

Vageli Coutsias

Problems

- Trigonometric Interpolation
- Gibbs phenomenon and aliasing
- The Fast Fourier Transform



Fourier series

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(kt) + b_k \sin(kt)$$

$$\begin{pmatrix} a_k \\ b_k \end{pmatrix} = \frac{1}{\pi} \int_{t=0}^{2\pi} f(t) \begin{pmatrix} \cos(kt) \\ \sin(kt) \end{pmatrix} dt$$

10/21/2002 3

$$f_n = a_0 + \sum_{k=1}^{m-1} \left(a_k \cos\left(\frac{kn\pi}{m}\right) + b_k \sin\left(\frac{kn\pi}{m}\right)\right) + (-1)^n a_m$$

$$n = 0, \cdots, 2m-1$$

$$a_k = \frac{1}{\pi} \int_{t=0}^{2\pi} f(t)\cos(kt)dt \approx \frac{1}{m} \sum_{n=0}^{2m-1} f_n \cos\left(\frac{kn\pi}{m}\right)$$

$$b_k = \frac{1}{\pi} \int_{t=0}^{2\pi} f(t)\sin(kt)dt \approx \frac{1}{m} \sum_{n=0}^{2m-1} f_n \sin\left(\frac{kn\pi}{m}\right)$$

10/21/2002 4

$C =$

$$\begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \cos\left(\frac{n\pi}{m}\right) & \cdots & (-1)^n & \sin\left(\frac{n\pi}{m}\right) & \cdots & \sin\left(\frac{n(m-1)\pi}{m}\right) \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \cos\left(\frac{(2m-1)\pi}{m}\right) & \cdots & (-1)^{2m-1} & \sin\left(\frac{(2m-1)\pi}{m}\right) & \cdots & \sin\left(\frac{n(2m-1)\pi}{m}\right) \end{pmatrix}$$

$$\hat{F} = \{a, b\} = \{[a_0, a_1, \cdots, a_m], [b_1, \cdots, b_{m-1}]\}^T$$

$$f = [f_0, f_1, \cdots, f_{2m-1}(, f_{2m} = f_0)]^T$$

10/21/2002

$$C\hat{F} = f \Leftrightarrow \hat{F} = C^{-1} f$$

5

```
function F =CSInterp(f)
n = length(f);
m = n/2;
tau = (pi/m)*(0:n-1)';
P = [];
for j = 0:m,     P = [P cos(j*tau)]; end
for j = 1:m-1,   P = [P sin(j*tau)]; end
y = P\f;
F = struct('a',y(1:m+1),'b',y(m+2:n));
```

$$f_n = a_0 + \sum_{k=1}^{m-1}\left(a_k \cos\left(\frac{k\pi}{m}\right) + b_k \sin\left(\frac{kn\pi}{m}\right)\right) + (-1)^n a_m$$

$n = 0, \cdots, 2m-1$

10/21/2002

6

THE FOURIER COEFFICIENTS

F

F.a=[y(1),…,y(m+1)]

F.b=[y(m+2),…,y(2m)]

A CELL array: it’s elements are STRUCTURES
(here arrays)

```
F = struct('a',y(1:m+1),'b',y(m+2:n));
```

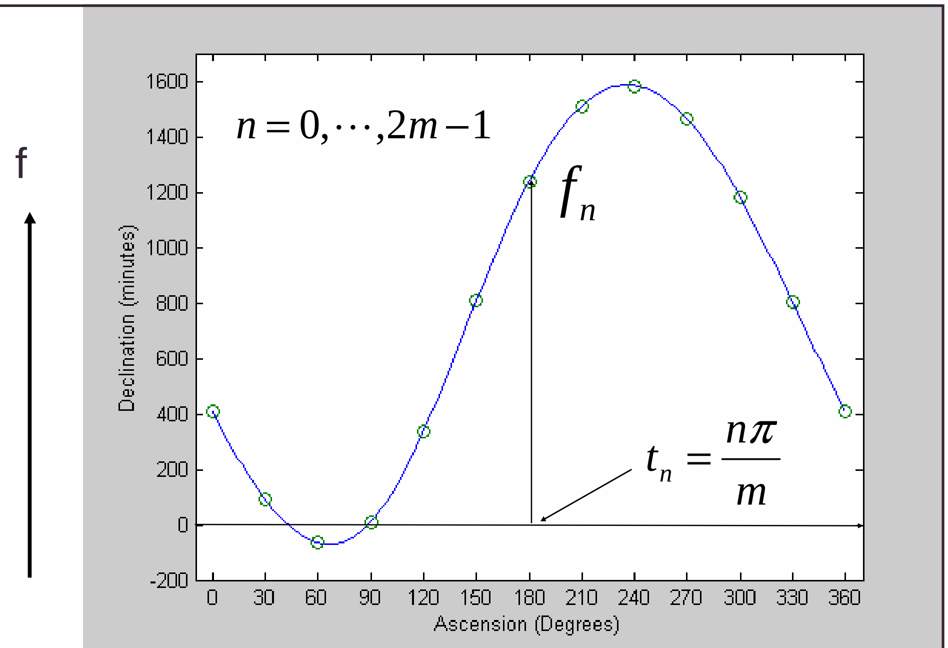
10/21/2002 7

```
function Fvals = CSeval(F,T,tvals)
        Fvals = zeros(length(tvals),1);
        tau = (2*pi/T)*tvals;
    for j = 0:length(F.a)-1
        Fvals = Fvals + F.a(j+1)*cos(j*tau);
    end
    for j = 1:length(F.b)
        Fvals = Fvals + F.b(j  )*sin(j*tau);
    end
```

$$f(t) = a_0 + \sum_{k=1}^{m-1}\left(a_k \cos\left(\frac{2\pi kt}{T}\right) + b_k \sin\left(\frac{2\pi kt}{T}\right)\right)$$

$$\cdots\cdots + a_m \cos\left(\frac{2\pi mt}{T}\right)$$

10/21/2002 8

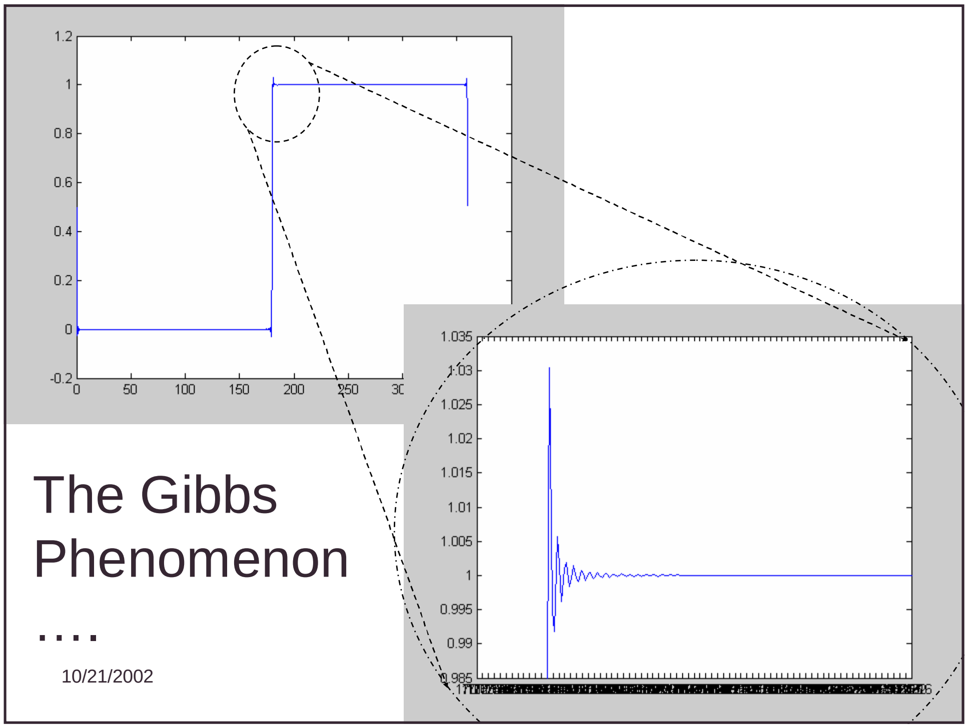


10/21/2002 9

t

```
% Script File: Pallas
A = linspace(0,360,13)';
D=[ 408 89 -66 10 338 807 1238 1511 1583 1462
1183 804 408]';
Avals = linspace(0,360,200)';
F     = CSInterp(D(1:12));
Fvals = CSeval(F,360,Avals);
plot(Avals,Fvals,A,D,'o')
axis([-10 370 -200 1700])
set(gca,'xTick',linspace(0,360,13))
xlabel('Ascension (Degrees)')
ylabel('Declination (minutes)')
```

10/21/2002 10



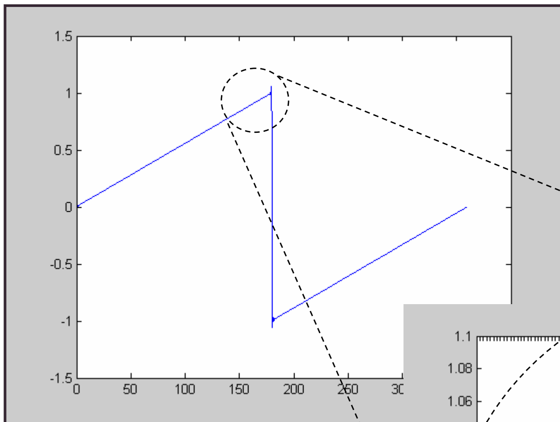
The Gibbs Phenomenon ….

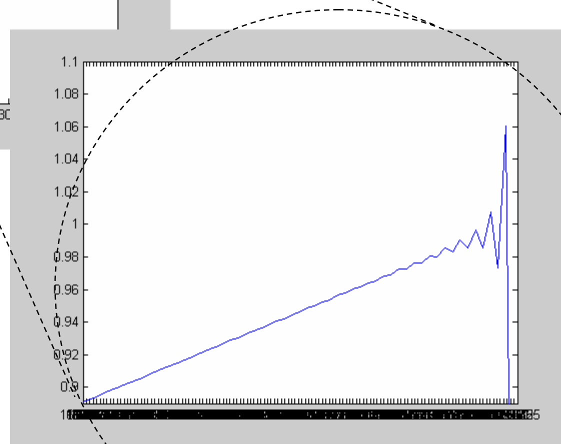
10/21/2002

% Script File: square
% Trigonometric interpolant of a square wave.

```
M=300
A = linspace(0,360,2*M+1)';
D(2:M)=0;D(M+2:2*M)=1;
D(1)=.5;D(M+1)=.5;D(2*M+1)=.5;
D=D';
Avals = linspace(0,360,1000)';
F     = CSInterp(D(1:2*M));
Fvals = CSeval(F,360,Avals);
plot(Avals,Fvals)
```

10/21/2002 12



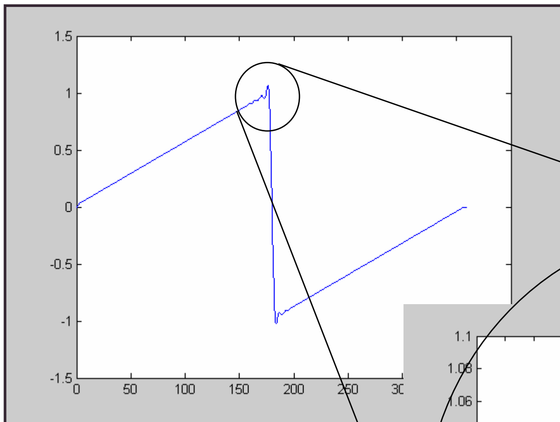


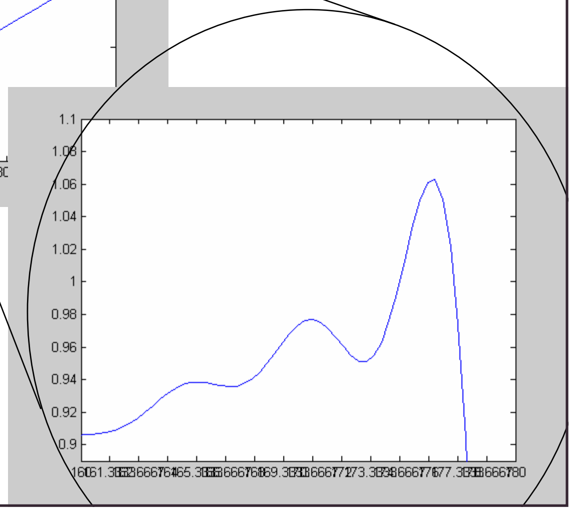
….
can be seen at
all discontinuities

10/21/2002

```
% Script File: sawtooth_wave.
clear
M=512
A = linspace(0,360,2*M+1)';
D(2:M)=(2:M)/M;D(M+2:2*M)=-1+(2:M)/M;
D(1)=0;D(M+1)=0;D(2*M+1)=0;
D=D';
Avals = linspace(0,360,1024)';
F     = CSInterp(D(1:2*M));
Fvals = CSeval(F,360,Avals);
plot(Avals,Fvals)
```

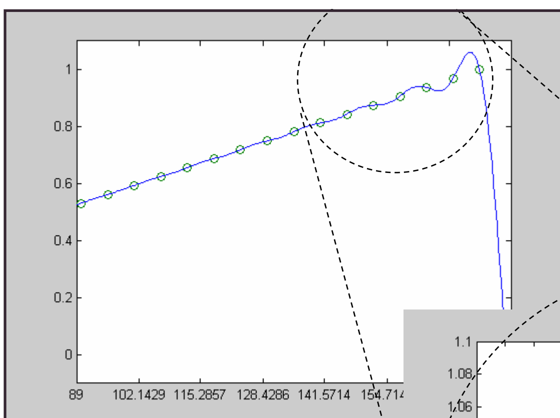
10/21/2002 14

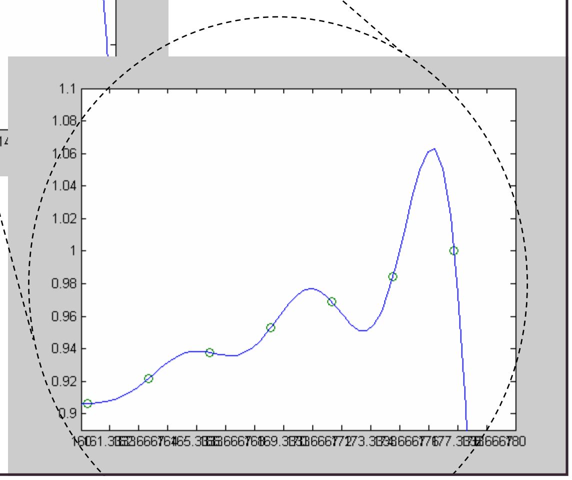




The overshoot is present at any resolution…

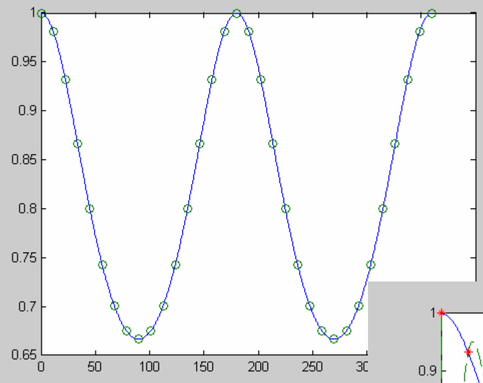
10/21/2002



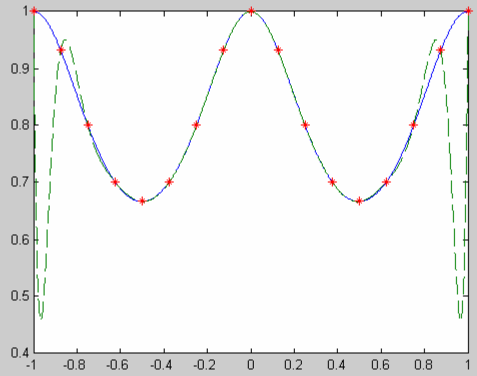


…although the interpolant is correct at the nodes as dx->0

10/21/2002



$$y = \frac{1}{1+.5\sin^2 \pi x},$$
$$-1 \le x \le 1$$



Compare with equispaced polynomial interpolation


```
clear
m=8;n=2*m;
       x = linspace(-1,1,n+1);
       y = 1./(1+.5*(sin(x*pi)).^2);
     x0 = linspace(-1,1,1000);
     y0 = 1./(1+.5*sin(x0*pi).^2);
     a = InterpNRecur(x,y);
pVal = HornerN(a,x,x0);
       plot(x0,y0,x0,pVal,'--',x,y,'*')
      axis([-1 1 0.4 1.2])
```

```
 function y = DFT(x,isign)
% y = DFT(x,+/-1):  +1 input:point sample   -
> output:freq. dom. coeffs.
%                    -1 input:freq.dom.coef. ->
output:point values
n = length(x);
y = x(1)*ones(n,1);
if n>1
   m=n/2;
   w=exp(isign*pi*sqrt(-1)/m);
   v = w.^(0:n-1)';
   for k = 2:n
```

10/21/2002 19

```
if n>1
   m=n/2;
   w=exp(isign*pi*sqrt(-1)/m);
   v = w.^(0:n-1)';
   for k = 2:n
      z = rem((k-1)*(0:n-1)',n )+1;
      y = y + v(z)*x(k);
   end
   if isign == 1
      y = y/m;
   end
end
```

10/21/2002 20

```
 function y = FFTRecur(x)      %n=2^k, x col.
n = length(x);
if n == 1
   y = x;
else
   m = n/2;
   yT = FFTRecur(x(1:2:n));
   yB = FFTRecur(x(2:2:n));
   d  = exp(-2*pi*sqrt(-1)/n).^(0:m-1)';
   z  = d.*yB;
   y  = [ yT+z ; yT-z ];
end
```

10/21/2002 21

FFTrecur is of order $n \log n$

DFT is of order n^2

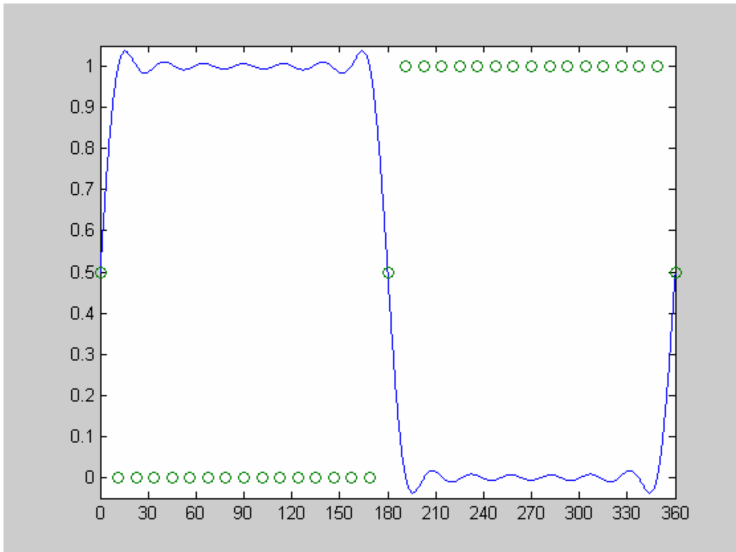
10/21/2002 22

```
% script heavyside-1
k = input('size of problem: (2*k) ');
m = 2^k; n = 2*m;
A = linspace(0,360,n+1);
x=zeros(n+1,1);
x(m+2:n) = 1;
x(m+1) = .5;
x(1) = .5; x(n+1) = .5;
y2 = FFTRecur(x(1:n));
```

10/21/2002 23

```
% script heavyside-2
P2 = struct('a',real(y2(1:m+1))/m,'b',
                    -imag(y2(2:m-1))/m);
P2.a(1)=P2.a(1)/2;P2.a(m+1)=P2.a(m+1)/2;
Bvals = linspace(0,360,2000)';
Gvals = CSeval(P2,360,Bvals);
plot(Bvals,Gvals,A,x,'o')
axis([0 360 -.05 1.05])
set(gca,'xTick',linspace(0,360,13))
```

10/21/2002 24



Hint: fix the “bug” and you have the secret to the FFT problem!

10/21/2002 25

Summary

- Trigonometric Interpolation
- Fourier Series
- Gibbs phenomenon
- Equispaced interpolation: polynomial vs trigonometric

References

- find m-files at
  www.math.unm.edu/~vageli
- Higham & Higham,
  Matlab Handbook
- C.F.van Loan,
  Intro to Sci.Comp. (2-3)