Hermite-Padé approximation

Here we approximate the function considered in Olver & Xu, 2020:

$$f(x) = \sin(10x + 20\sqrt{x^2+\epsilon^2}), \qquad x \in [-1, 1], \qquad \epsilon = 0.01. \qquad (\texttt{example})$$

Since this function has square root-type branch points, we might try Hermite-Padé (HP) approximation because HP approximants have algebraic branch points.

Let

$$\langle f, g\rangle = \sum_{k=0}^{N} w_k f(x_k) g(x_k), \qquad x_k = \cos(k\pi/N), \qquad w_0 = w_N = \frac{1}{2}, w_k = 1, k \neq 1, N,$$

and

$$p_j(x) = \sum_{k=0}^{d_j} \sqrt{w_k}\, c_k T_k(x), \qquad w_0 = w_{d_j} = \frac{1}{2}, w_k = 1, k \neq 0, d_j,$$

then

$$\|p_j\|^2 = \langle p_j, p_j\rangle = \|\mathbf{c}\|_2^2 = |c_0|^2 + \cdots + |c_{d_j}|^2. \qquad (\texttt{iso})$$

Suppose we have the function values $f(x_k)$, $k = 0, \ldots, N$. We want to find polynomials $p_0, \ldots, p_m$ of degrees $d_0, \ldots, d_m$ such that

$$\|p_0 + p_1 f + \cdots + p_m f^m\| = \text{minimum}, \qquad (\texttt{LS})$$

We assume some kind of normalization so that the trivial solution $p_0 = \cdots = p_m = 0$ is not admissable. Because of the isometry (`iso`), (`LS`) is a least squares problem whose solution can be computed with the SVD. If the number of unknown polynomial coefficients matches the number of points on the Chebyshev grid, we obtain the 'interpolation' case:

$$n := \sum_{j=0}^{m} d_j + m, \qquad N + 1 = n \qquad \Rightarrow \qquad \|p_0 + p_1 f + \cdots + p_m f^m\| = \text{minimum} = 0. \qquad (\texttt{interp})$$

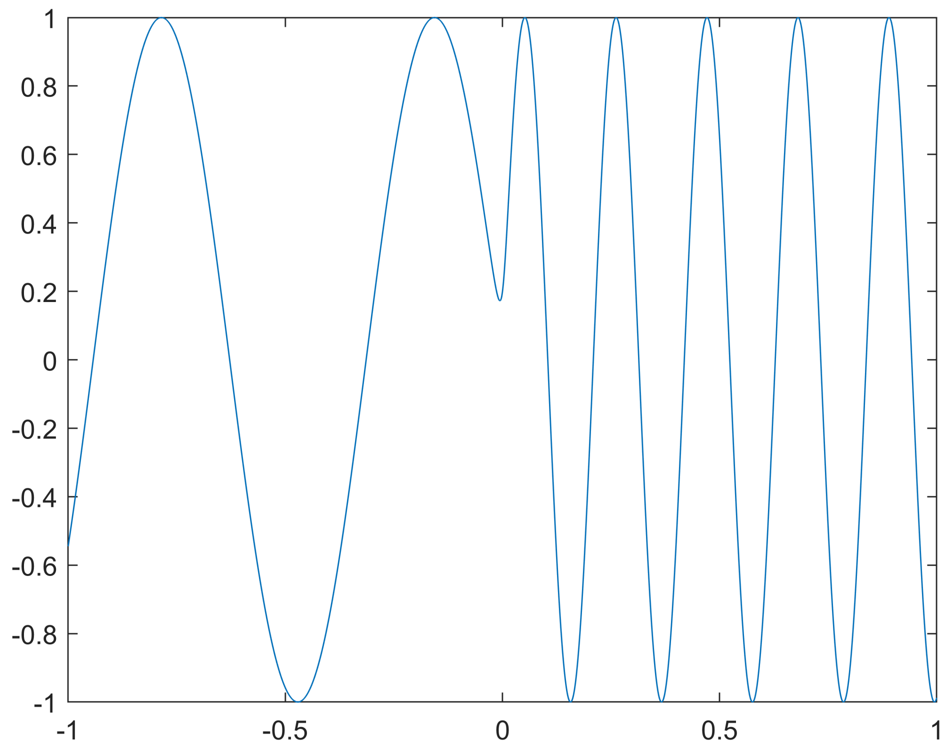
The Hermite-Padé approximant of $f(x)$, viz. $\psi(x)$, is the algebraic function defined by

$$p_0(x) + p_1(x)\psi(x) + \cdots + p_m(x)\psi^m(x) = 0. \qquad (\texttt{HPeq})$$

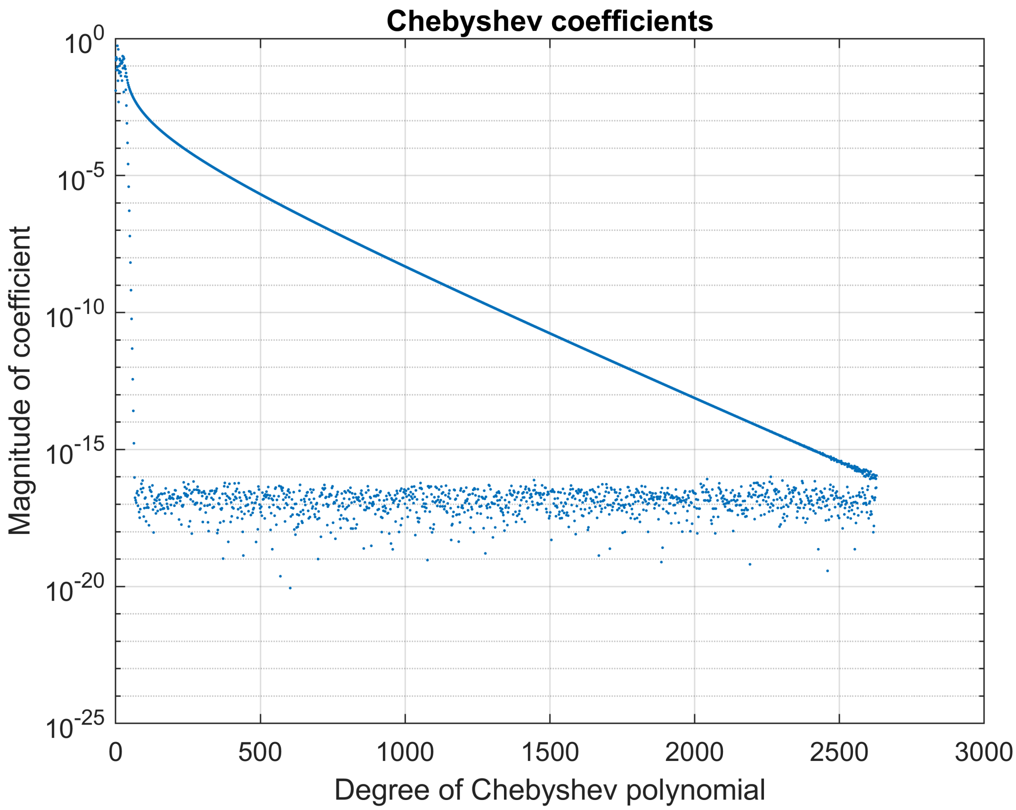
Note that if $m = 0$ and $p_1(x) = 1$ in (`interp`), then the HP approximant, $\psi(x) = -p_0(x)$, is a polynomial interpolant of*f* on the grid; if $m = 1$, then the HP approximant, $\psi(x) = -p_0(x)/p_1(x)$, is a rational interpolant of *f*(with poles in the complex x-plane). If $m \geq 2$, then for every x, $\psi(x)$ will generally be an m-valued approximant of *f*(with poles and algebraic branch points in the complex x-plane). We want to pick only one branch of the m-valued function ψ to approximate *f*. One way to do this is to solve (`HPeq`) with Newton's method using a polynomial or rational approximant as first guess. Here we will 'cheat' in the case $m = 2$ by using the quadratic formula to solve (`HPeq`) and then picking the solution/branch that is closest to $f(x)$.

First, consider a polynomial approximation to (`example`):

```
f = @(t) sin(10*t + 20*sqrt(t.^2 + epsilon^2));
fc = chebfun(f);
plot(fc)
```

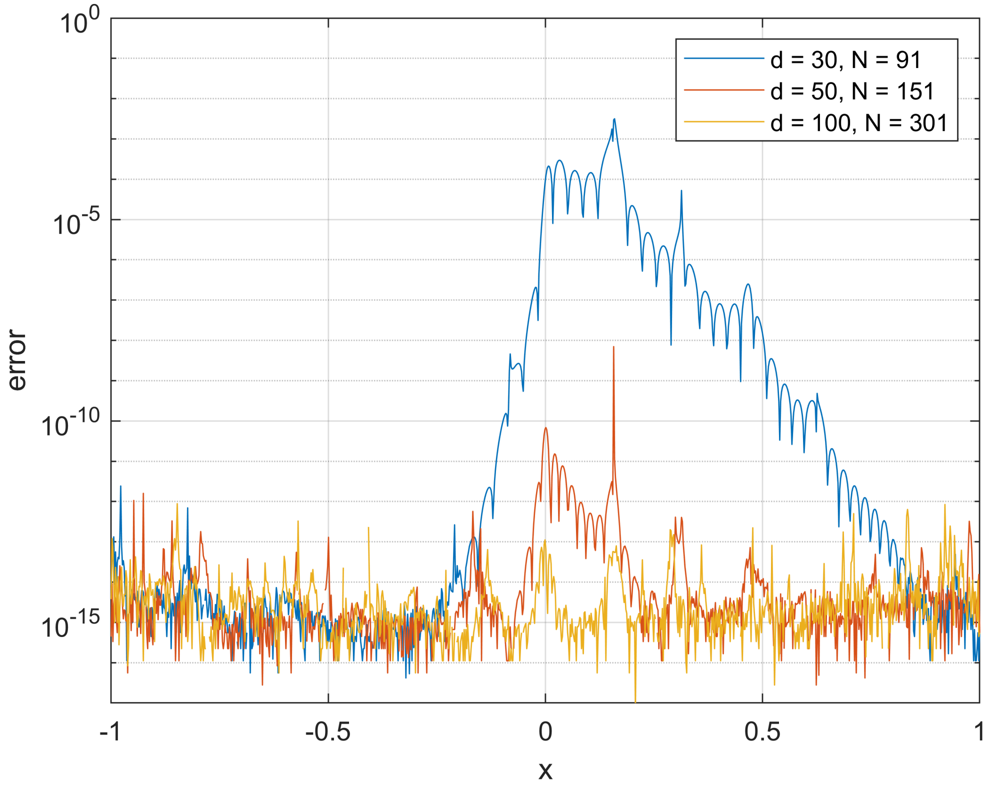



```
plotcoeffs(fc)
```



Examples of quadratic (i.e., $m = 2$) Hermite-Padé approximants for the 'interpolation' case. Let the degrees of the polynomials be equal, $d_0 = d_1 = d_2 = d$, then the number of points on the grid is $N + 1 = 3d + 2$.

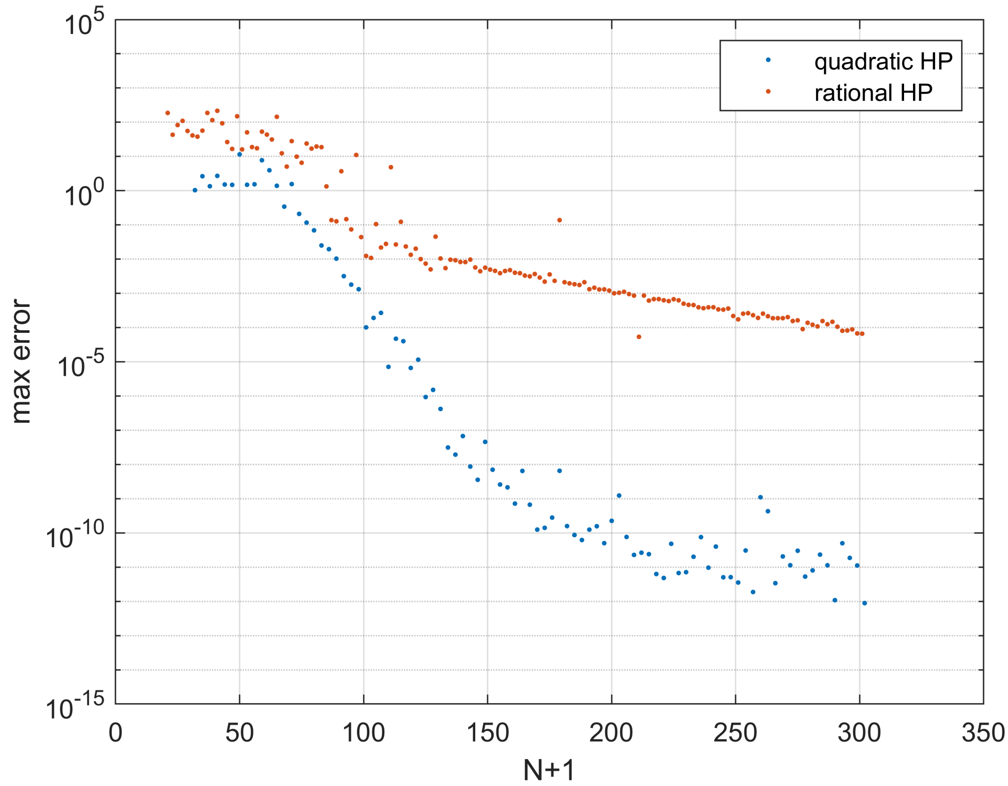
```
xx = linspace(-1,1,1000)';
d = 30;
fff = f(xx);
psi = quadHPapproxinterp(f,xx,fff,d);
semilogy(xx,abs(psi-fff)), grid on, hold on
d = 50;
psi = quadHPapproxinterp(f,xx,fff,d);
semilogy(xx,abs(psi-fff))
d = 100;
psi = quadHPapproxinterp(f,xx,fff,d);
semilogy(xx,abs(psi-fff)),axis([-1 1 1e-17 1e0])
legend('d = 30, N = 91','d = 50, N = 151','d = 100, N = 301')
xlabel('x')
ylabel('error')
hold off
```



Compare the rate of convergence of quadratic and rational ($m = 1$) Hermite-Padé approximants in the interpolation case as a function of the number of grid points, $N + 1$. As before, let the polynomials have equal degrees.

```
dvec = 10:100;
quaderror = zeros(length(dvec),1);
Np1 = 3*dvec+2;
for d = dvec
    psi = quadHPapproxinterp(f,xx,fff,d);
    quaderror(d-dvec(1)+1) = max(abs(psi - fff));
end
semilogy(Np1,quaderror,'.'), grid on, hold on

dvec = 10:round(1.5*dvec(end));
raterror = zeros(length(dvec),1);
Np1 = 2*dvec+1;
for d = dvec
    psi = ratHPapproxinterp(f,xx,d);
    raterror(d-dvec(1)+1) = max(abs(psi - fff));
end
semilogy(Np1,raterror,'.'), hold off
legend('quadratic HP','rational HP')
xlabel('N+1')
ylabel('max error')
```




```
function psi = quadHPapproxinterp(f,xx,fff,d)
% compute a quadratic HP approximant in the 'interpolation' case. Let all
% the polynomials have the same degree
n = 3*d + 2;
N = n-1;
x = chebpts(N+1);
ff = f(flipud(x));
psi = zeros(length(fff),1);
tol = 0;
degs = d*[1, 1, 1];
coeffs = MVLS_Cheb_2nd_kind3(ff,degs,tol);
evals = MVLS_Cheb_2nd_kind_eval(coeffs,degs,xx);
p0 = evals(:,:,1);
p1 = evals(:,:,2);
p2 = evals(:,:,3);
% solve p0 + p1*psi + p2*psi^2 = 0
psi1 = (-p1 + sqrt(p1.^2 - 4*p0.*p2))./(2*p2);
psi2 = (-p1 - sqrt(p1.^2 - 4*p0.*p2))./(2*p2);
e1 = abs(fff - psi1);
e2 = abs(fff - psi2);
inds1 = e1<e2;
psi(inds1) = psi1(inds1);
psi(~inds1) = psi2(~inds1);
end

function psi = ratHPapproxinterp(f,xx,d)
% compute a diagonal rational interpolant on the Chebyshev grid, hence the
```

```
% numerator and denominator polynomials have the same degree
n = 2*d + 1;
N = n-1;
x = chebpts(N+1);
ff = f(flipud(x));
tol = 0;
degs = d*[1, 1];
coeffs = MVLS_Cheb_2nd_kind3(ff,degs,tol);
evals = MVLS_Cheb_2nd_kind_eval(coeffs,degs,xx);
p0 = evals(:,:,1);
p1 = evals(:,:,2);
% solve p0 + p1*psi = 0
psi = -p0./p1;
end
```