Maple worksheet by E.S. Phinney 3/9/2004 ACM95b/100b handout

Numerical solution to the Sturm-Liouville problem

$$\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + qy + \lambda w(x)y = 0 \tag{1}$$

with $y(0) = 0$, $y(1) = 0$, $p(x) = 1$, $q(x) = 0$, i.e. $y'' + \lambda y = 0$ (so exact eigenvalues are $\lambda_m = m^2\pi^2$ and eigenfunctions $\phi_m(x) = \sqrt{2}\sin(m\pi x)$, $m = 1, 2, 3, \ldots$.

We do this by finding Rayleigh-Ritz minimum of

$$R = \frac{\int_a^b [p(x)(y')^2 - y^2 q(x)]\, dx}{\int_a^b y^2 w(x) dx} \tag{2}$$

or equivalently, the minimum of the numerator subject to the condition that the denominator be equal to 1. To do this numerically, we use the Euler polygonal approximation to $y(x)$ and its derivatives and the integral. This is a simple form of the *Finite Element Method*, for finding a simple *Finite Difference* approximation eq (6) to the ODE boundary value problem. Defining $\Delta x = 1/n$ (number of grid points into which we have divided the interval [0,1]), we want to minimize

$$\begin{aligned} R &= \sum_{j=0}^{n-1} (y'_{j+1/2})^2 \Delta x && (3) \\ &= \sum_{j=0}^{n-1} (y_{j+1} - y_j)^2 / \Delta x;, && (4) \\ \text{subject to } 1 &= \sum_{j=1}^{n-1} y_j^2 \Delta x && (5) \end{aligned}$$

Write eq (4) out, and notice that for each k, y_k appears in three terms (except at the boundaries where it appears in two). The Rayleigh-Ritz extremum principle says that we want to minimize R over all possible choices of y_k. To take account of the constraint, which makes one of the y_j be determined by all the others, we introduce a Lagrange multiplier λ to uncouple it (just as we did in the continuous case). Then taking $\partial/\partial y_k$ of the expression for R in eq (4) and setting it to zero (Rayleigh-Ritz extremum) for each k, with boundary conditions $y_0 = 0$ and $y_n = 0$ gives

$$y_{j+1} - 2y_j + y_{j-1} + \lambda \Delta x^2 y_j = 0 \tag{6}$$

This can be written as a band diagonal matrix (with band elements $[1, -2 + \lambda\Delta x^2, 1]$ as shown below) times the column matrix $(y_1, y_2, \ldots, y_{n-1})$. We compute numerical approximations to the eigenvalues λ by evaluating the determinant of this matrix for $n = 5$ (matrix A), $n = 10$ (matrix B) and $n = 30$ (matrix C) below. We could also the solve for the y_j for each eigenvalue, and get the polygonal approximations to the eigenfunctions as well.

In the real world, one doesn't use symbolic Maple for this sort of calculation, but fast matrix solvers, specialised to sparse bands like this.

```
> with(linalg):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected
```

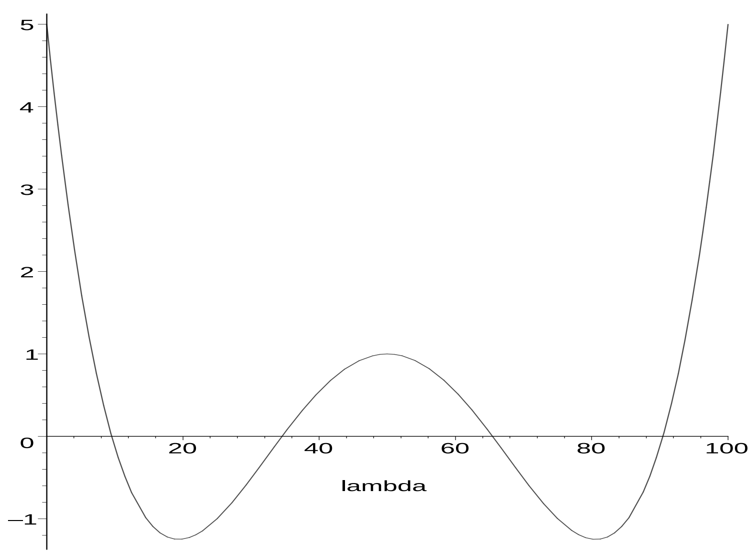
```
> A:= band([1,-2+lambda/25,1],4);
```

$$A := \begin{bmatrix} -2+\frac{1}{25}\lambda & 1 & 0 & 0 \\ 1 & -2+\frac{1}{25}\lambda & 1 & 0 \\ 0 & 1 & -2+\frac{1}{25}\lambda & 1 \\ 0 & 0 & 1 & -2+\frac{1}{25}\lambda \end{bmatrix}$$

```
> detofa := det(A);
```

$$detofa := 5 - \frac{4}{5}\lambda + \frac{21}{625}\lambda^2 - \frac{8}{15625}\lambda^3 + \frac{1}{390625}\lambda^4$$

```
> plot(detofa,lambda=0..100);
```




```
> fsolve(detofa=0);
```

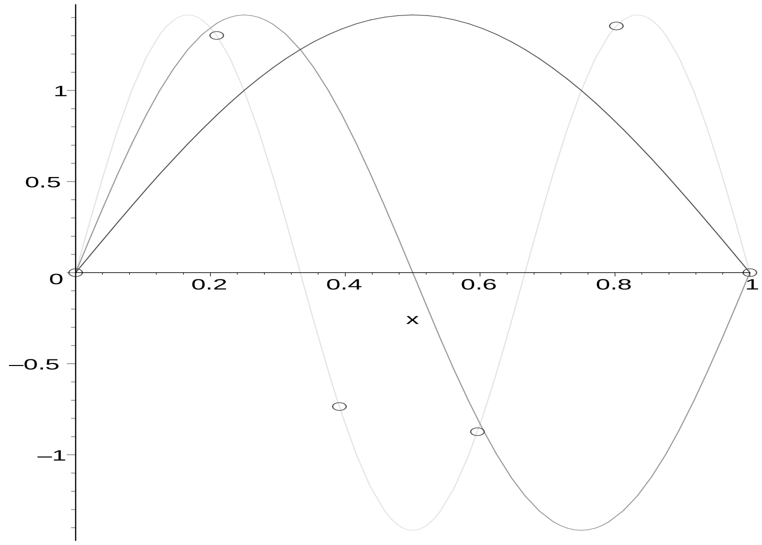
$$9.549150281, 34.54915028, 65.45084972, 90.45084972$$

```
> exacteigen:= [seq(j^2*Pi^2,j=1..3)];
> evalf(exacteigen);
```

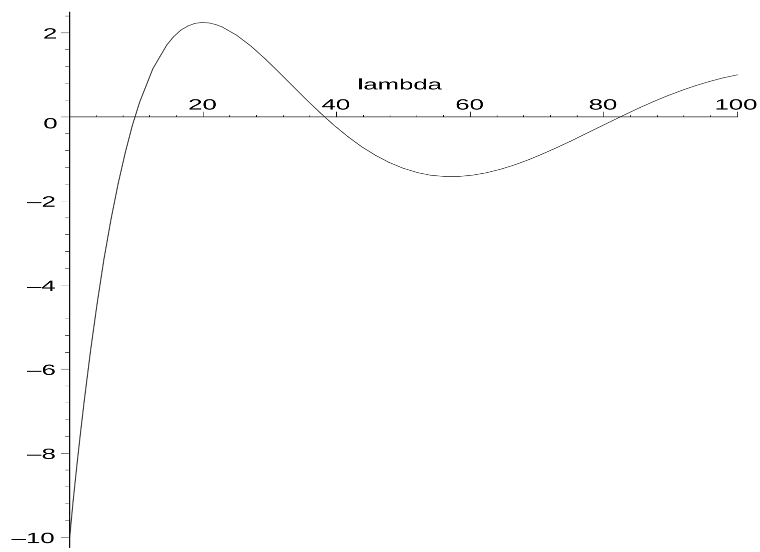
$$exacteigen := [\pi^2, 4\pi^2, 9\pi^2]$$

$$[9.869604404, 39.47841762, 88.82643964]$$

Comparing with the numerical eigenvalues found from the 5-grid point (4 nonzero grid points) approximation with the 5×5 matrix above, we see that we made a 3% error in the first eigenvalue, 13% on the second and 27% on the third. The bad performance on the last is hardly surprising: we are trying to approximate a function with two nodes with only four numbers (circles in plot of first 3 eigenfunctions below)!




```
> B:= band([1,-2+lambda/100,1],9):
> detofb := det(B):
> plot(detofb,lambda=0..100);
```



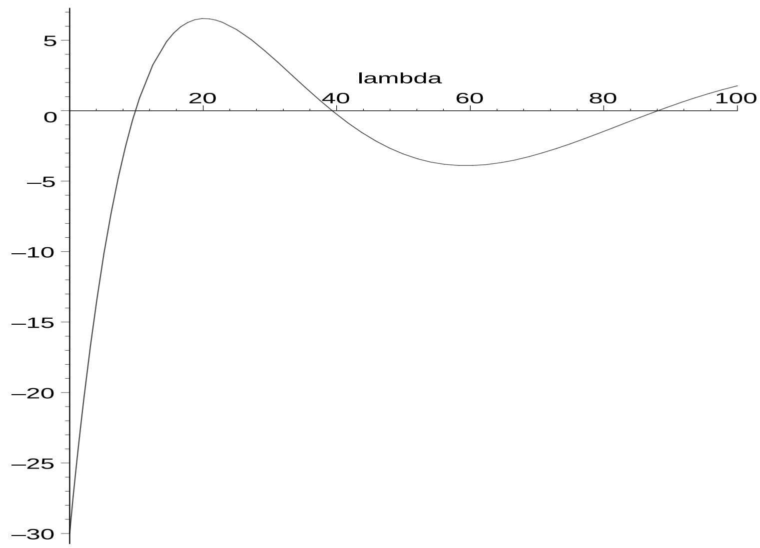
```
> fsolve(detofb=0,lambda=0..100);
```

9.788696741, 38.19660113, 82.44294954

```
> evalf(exacteigen);
```

[9.869604404, 39.47841762, 88.82643964]

```
> C:=band([1,-2+lambda/900,1],29):
> detofc :=det(C):
> plot(detofc,lambda=0..100);
```




```
> fsolve(detofc=0,lambda=0..100);
```

9.860588337, 39.33431868, 88.09827067

```
> evalf(exacteigen);
```

[9.869604404, 39.47841762, 88.82643964]