Lecture 21

Numerical differentiation

1 Introduction

We can analytically calculate the derivative of any elementary function, so there might seem to be no motivation for calculating derivatives numerically. However we may need to estimate the derivative of a numerical function, or we may only have a fixed set of sampled function values. In these cases we need to estimate the derivative numerically.

2 Finite difference approximation

The definition of the derivative of a function $f(x)$ that you will most often find in calculus textbooks is

$$\frac{df}{dx}=\lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \tag{1}$$

This immediately suggests the approximation

$$\frac{df}{dx}\approx\frac{\Delta f}{\Delta x}=\frac{f(x+h)-f(x)}{h} \tag{2}$$

where the *step size h* is small but not zero. This is called a *finite difference*. Specifically it's a *forward difference* because we compare the function value at x with its value at a point "forward" of this along the *x* axis, $x+h$.

How small should *h* be? Because of round-off error, smaller is not always better. Let's use Scilab to estimate

$$\left.\frac{d}{dx}e^x\right|_{x=0}=1\approx\frac{e^h-e^0}{h} \tag{3}$$

for various *h* values. The absolute error in the estimate vs. *h* is graphed in Fig. 1. As *h* decreases from 10^{-1} down to 10^{-8} the error decreases also. However for $h=10^{-9}$ and smaller the error actually increases! The culprit is round-off error in the form of "the small difference of large numbers." Double precision arithmetic provides about 16 digits of precision. If $h\approx 10^{-16}$ then $e^h\approx e^0$ to 16 digits and the difference e^h-e^0 will be very inaccurate. When $h=10^{-8}$ the difference e^h-e^0 will be accurate to about 8 digits or about 10^{-8} , the point at which theoretical improvement in numerical accuracy is offset by higher round-off error. We typically ignore round-off error when estimating numerical accuracy, but round-off error needs to be kept in mind when implementing any algorithm.

Let's investigate how the numerical accuracy of our estimate varies with step size *h*. Assume we want to estimate the derivative of $f(x)$ at $x=0$. Write *f* as a power series

$$f(x)=f(0)+\sum_{n=1}^{\infty}\frac{1}{n!}f^{(n)}(0)x^n \tag{4}$$

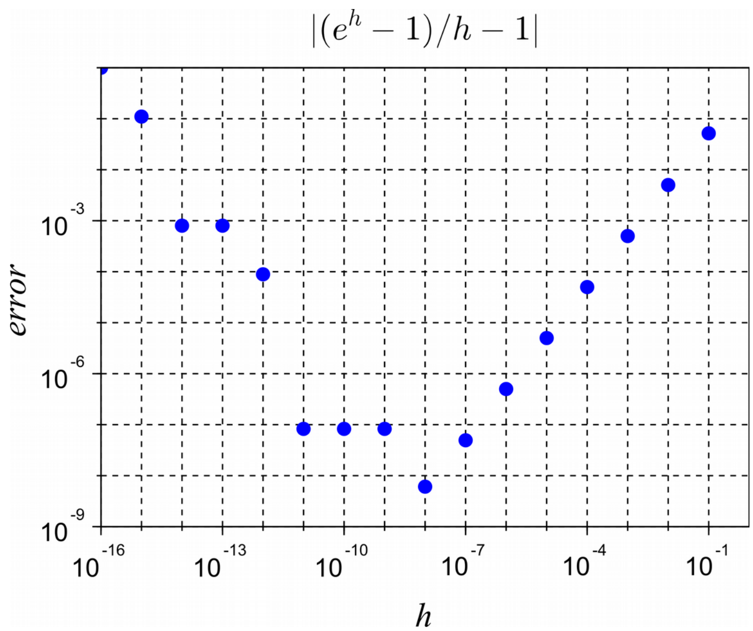

Fig. 1: Error in forward-difference estimation of derivative of e^x *at* $x=0$ *vs. step size h. As h decreases from* 10^{-1} *to* 10^{-8}*, numerical error decreases proportionally. As h decreases further, round-off error begins to exceed numerical error.*

Then

$$\frac{f(h)-f(0)}{h}=\sum_{n=1}^{\infty}\frac{1}{n!}f^{(n)}(0)h^{n-1}=f'(0)+\frac{1}{2}f''(0)h+\frac{1}{6}f'''(0)h^2+\cdots \tag{5}$$

For small *h* therefore

$$\frac{f(h)-f(0)}{h}=f'(0)+\frac{1}{2}f''(0)h+\cdots \tag{6}$$

and we say that the approximation is *first-order accurate* since the error (the second term on the right) varies as the first power of *h*. Decreasing *h* by a factor $1/10$ will decrease the error by $1/10$. However, as we see in Fig. 1 this is only true up to the point that round-off error begins to be significant. For double precision $h \approx 10^{-8}$ is optimal. We can shift (6) along the *x* axis and rearrange to obtain the general forward-difference formula

$$f'(x)=\frac{f(x+h)-f(x)}{h}+O(h) \tag{7}$$

3 Higher-order formulas

The forward-difference approximation (6) uses two samples of the function, namely $f(0)$ and $f(h)$. Using three samples we might be able to get a better estimate of $f'(0)$. Suppose we have the samples $f(-h), f(0), f(h)$. In terms of the power series representation of the function these are

$$\begin{aligned} f(-h) &= f(0)-f'(0)h+\frac{1}{2}f''(0)h^2-\frac{1}{3!}f^{(3)}(0)h^3+\frac{1}{4!}f^{(4)}(0)h^4-\frac{1}{5!}f^{(5)}(0)h^5+\cdots \\ f(0) &= f(0) \\ f(h) &= f(0)+f'(0)h+\frac{1}{2}f''(0)h^2+\frac{1}{3!}f^{(3)}(0)h^3+\frac{1}{4!}f^{(4)}(0)h^4+\frac{1}{5!}f^{(5)}(0)h^5+\cdots \end{aligned} \tag{8}$$

Let's use a linear combination of these values to estimate $f'(0)$ as

$$f'(0)\approx a\,f(-h)+b\,f(0)+c\,f(h) \tag{9}$$

where a,b,c are unknown coefficients that we will choose to get the best possible estimate. The sum $a\,f(-h)+b\,f(0)+c\,f(h)$ will include a term with a factor of $f(0)$. We want this to vanish. This requires

$$(a+b+c)\,f(0)=0\rightarrow a+b+c=0 \tag{10}$$

which is one equation in three unknowns. Terms with a factor of $f'(0)$ should combine to give the derivative value $f'(0)$. This requires

$$(-a+c)\,f'(0)h=f'(0)\rightarrow -a+c=\frac{1}{h} \tag{11}$$

We now have two equations in three unknowns. To get a third equation we can require the next term, which contains a factor of $f''(0)h^2$, to vanish. This gives us the equation

$$(a+c)\frac{1}{2}f''(0)h^2=0\rightarrow a+c=0 \tag{12}$$

Our three equations in three unknowns can be written as

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \end{pmatrix}=\begin{pmatrix} 0 \\ \frac{1}{h} \\ 0 \end{pmatrix}$$

The solution is

$$a=-\frac{1}{2h},\ b=0,\ c=\frac{1}{2h}$$

and our approximation reads

$$f'(0)\approx\frac{f(h)-f(-h)}{2h}$$

Using (8) we have

$$\frac{f(h)-f(-h)}{2h}=f'(0)+\frac{1}{6}f^{(3)}(0)h^2+\cdots$$

so this approximation is *second-order accurate*. Decreasing h by a factor of $1/10$ should decrease the numerical error by a factor of $1/100$. Rearranging and writing this for an arbitrary value of x we have the formula

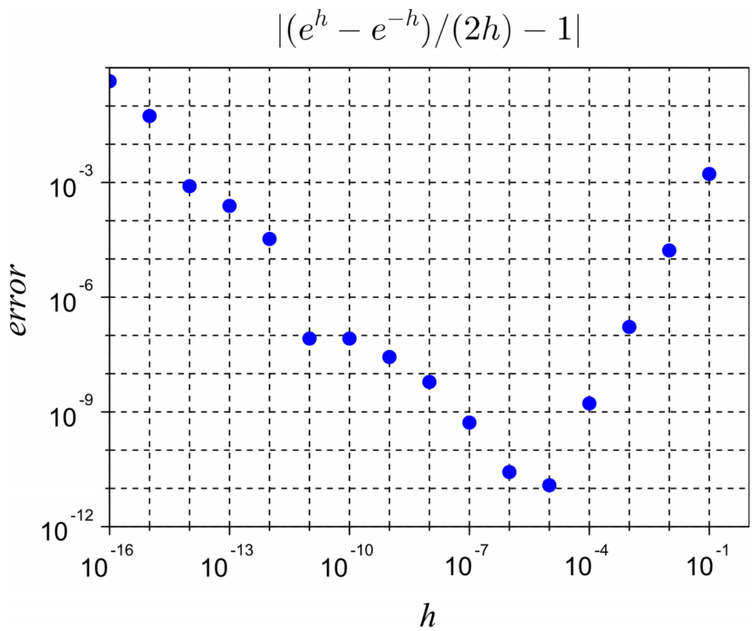

Fig. 2: Error in central-difference estimate of derivative of e^x *vs. h.*

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{13}$$

This type of finite difference is called a *central difference* since it uses both the forward sample $f(x+h)$ and the backward sample $f(x-h)$. Scilab code is given in the Appendix.

The error in the central-difference approximation

$$\left.\frac{d}{dx} e^x\right|_{x=0} = 1 \approx \frac{e^h - e^{-h}}{2h}$$

is plotted in Fig. 2. Note how the error reduces more rapidly with decreasing *h*. This allows the approximation to reach a greater accuracy before round-off error starts to become significant. With $h = 10^{-5}$ the error is only about 10^{-11}.

We extend this idea by using even more function samples. If we have the five samples $f(-2h), f(-h), f(0), f(h), f(2h)$ we can form an estimate

$$f'(0) \approx a\, f(-2h) + b\, f(-h) + c\, f(0) + d\, f(h) + e\, f(2h) \tag{14}$$

This has five unknowns, so we need to form five equations. In terms of the Taylor series representation of $f(x)$ our five samples have the form

$$
\begin{aligned}
f(2h) &= f(0)-2f'(0)h+2f''(0)h^2-\frac{8}{3!}f^{(3)}(0)h^3+\frac{16}{4!}f^{(4)}(0)h^4-\frac{32}{5!}f^{(5)}(0)h^5+\cdots \\
f(-h) &= f(0)-f'(0)h+\frac{1}{2}f''(0)h^2-\frac{1}{3!}f^{(3)}(0)h^3+\frac{1}{4!}f^{(4)}(0)h^4-\frac{1}{5!}f^{(5)}(0)h^5+\cdots \\
f(0) &= f(0) \\
f(h) &= f(0)+f'(0)h+\frac{1}{2}f''(0)h^2+\frac{1}{3!}f^{(3)}(0)h^3+\frac{1}{4!}f^{(4)}(0)h^4+\frac{1}{5!}f^{(5)}(0)h^5+\cdots \\
f(2h) &= f(0)+2f'(0)h+2f''(0)h^2+\frac{8}{3!}f^{(3)}(0)h^3+\frac{16}{4!}f^{(4)}(0)h^4+\frac{32}{5!}f^{(5)}(0)h^5+\cdots
\end{aligned}
\tag{15}
$$

To get the $f(0)$ terms in (14) to vanish requires

$$(a+b+c+d+e)f(0)=0\rightarrow a+b+c+d+e=0 \tag{16}$$

To get the $f'(0)$ terms to produce the value $f'(0)$ requires

$$(-2a-b+d+2e)f'(0)h=f'(0)\rightarrow -2a+b+d+2e=\frac{1}{h} \tag{17}$$

The remaining three equations are obtained by requiring the $f''(0)$, $f^{(3)}(0)$ and $f^{(4)}(0)$ terms to vanish:

$$\left(2a+\frac{b}{2}+\frac{d}{2}+2e\right)f''(0)h^2=0\rightarrow 4a+b+d+4e=0$$

$$\frac{1}{3!}(-8a-b+d+8e)f^{(3)}(0)h^3=0\rightarrow -8a-b+d+8e=0$$

$$\frac{1}{4!}(16a+b+d+16e)f^{(4)}(0)h^4=0\rightarrow 16a+b+d+16e=0$$

$$\frac{1}{4!}(16+b+d+16e)f^{(4)}(0)h^4=0\rightarrow 16a+b+d+16e=0$$

Our five equations in five unknowns form the system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 1 & 4 \\ -8 & -1 & 0 & 1 & 8 \\ 16 & 1 & 0 & 1 & 16 \end{pmatrix}\begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix}=\begin{pmatrix} 0 \\ \frac{1}{h} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{18}$$

which has the solution

$$a=\frac{1}{12h},\ b=-\frac{8}{12h},\ c=0,\ d=\frac{8}{12h},\ e=-\frac{1}{12h} \tag{19}$$

Our approximation is therefore

$$f'(0)\approx\frac{f(-2h)-8f(-h)+8f(h)-f(2h)}{12h} \tag{20}$$

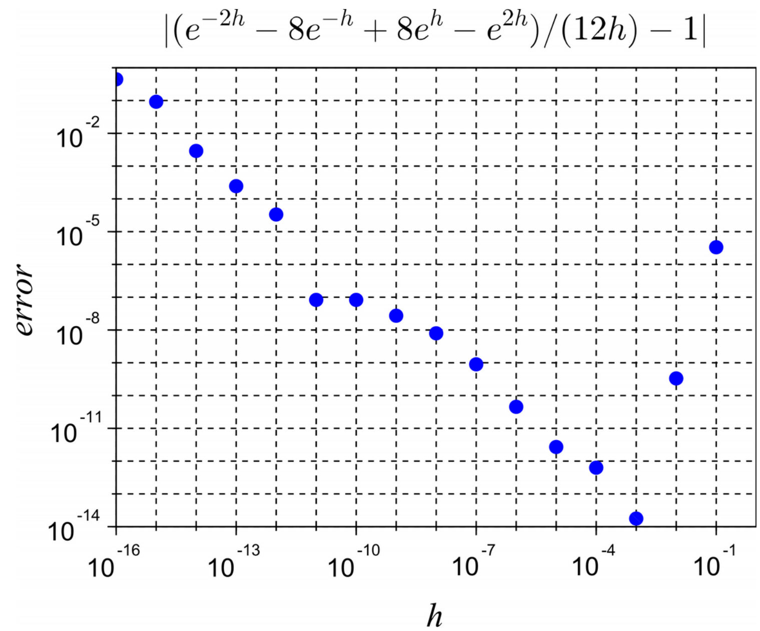

Fig. 3: Fourth-order accurate central-difference approximation to the derivative of e^x .

Using (15) the $f^{(5)}(0)$ terms are

$$f^{(5)}(0)\frac{h^5}{12h\cdot 5!}(-32+8+8-32)=-\frac{1}{30}f^{(5)}(0)h^4 \tag{21}$$

so

$$\frac{f(-2h)-8f(-h)+8f(h)-f(2h)}{12h}=f'(0)-\frac{1}{30}f^{(5)}(0)h^4+\cdots \tag{22}$$

We see that this is a *fourth-order accurate* approximation. Decreasing *h* by a factor of $1/10$ should decrease the numerical error by a factor of $1/10{,}000$. This is illustrated in Fig. 3 for $f(x)=e^x$. For $h=10^{-3}$ the error is only about 10^{-14} .

For arbitrary *x* value our *fourth-order central difference* approximation is

$$f'(x)=\frac{f(x-2h)-8f(x-h)+8f(x+h)-f(x+2h)}{12h}+O(h^4) \tag{23}$$

Scilab code is given in the Appendix.

4 The numderivative function (Scilab)

Scilab has a numerical derivative function named `numderivative`. To calculate the derivative of the function $f(x)$ at $x=x_0$ we execute

```
fp = numderivative(f,x0);
```

If *f* is a function several variables $f(\mathbf{x})$, then `numderivative` will return the gradient of *f*. It is also possible to specify the step size *h* and the order of the approximation (1,2 or 4).

```
fp = numderivative(f,x0,h,order);
```

The default is second order (central difference) and Scilab chooses an “optimal” value of *h*.

Some examples:

```
-->deff('y=f(x)','y=exp(-x)');

-->numderivative(f,1) //default central difference with optimal h
 ans  =
  - 0.3678794  //exact value is -exp(-1)=-0.3678794...

-->(f(1.1)-f(1))/0.1 //forward difference h=0.1
 ans  =
  - 0.3500836

-->numderivative(f,1,0.1,1) //forward difference h=0.1
 ans  =
  - 0.3500836

-->(f(1.1)-f(0.9))/0.2 //central difference h=0.1
 ans  =
  - 0.3684929

-->numderivative(f,1,0.1,2) //central difference h=0.1
 ans  =
  - 0.3684929
```

In most cases the default suffices.

5 Second derivative

With reference to (8), if we want to approximate the second derivative $f''(0)$ as

$$f''(0) \approx a\,f(-h) + b\,f(0) + c\,f(h) \tag{24}$$

we would require

$$(a+b+c)\,f(0) = 0 \to a+b+c = 0 \tag{25}$$

$$(-a+c)\,f'(0)\,h = 0 \to -a+c = 0 \tag{26}$$

and

$$(a+c)\frac{1}{2}f''(0)\,h^2 = f''(0) \to a+c = \frac{2}{h^2} \tag{27}$$

The solution to these three equations is

$$a = c = \frac{1}{h^2},\, b = -\frac{2}{h^2} \tag{28}$$

and we find

$$\frac{f(h) - 2\,f(0) + f(-h)}{h^2} = f''(0) + \frac{1}{12}f^{(4)}(0)\,h^2 + \cdots \tag{29}$$

so the approximation is second-order accurate. For arbitrary *x* value we have

$$f''(x)=\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+O(h^2) \tag{30}$$

6 Partial derivatives

For a function $f(x,y)$ the partial derivative $\frac{\partial f}{\partial x}$ can be defined as

$$\frac{\partial f}{\partial x}=\lim_{h\to 0}\frac{f(x+h,y)-f(x,y)}{h}$$

that is, we hold *y* fixed and compute the derivative as if *f* was only a function of *x*. A central-difference approximation is

$$\frac{\partial f}{\partial x}\approx\frac{f(x+h,y)-f(x-h,y)}{2h} \tag{31}$$

Likewise

$$\frac{\partial f}{\partial y}\approx\frac{f(x,y+h)-f(x,y-h)}{2h} \tag{32}$$

and

$$\frac{\partial^2 f}{\partial x^2}\approx\frac{f(x+h,y)-2f(x,y)+f(x-h,y)}{h^2} \tag{33}$$

Mixed partial derivative approximations such as $\frac{\partial^2 f}{\partial y\partial x}$ can be developed in steps such as

$$\begin{aligned}\frac{\partial^2 f}{\partial y\partial x}&\approx\frac{\left[\frac{\partial f}{\partial x}\right]_{y+h}-\left[\frac{\partial f}{\partial x}\right]_{y-h}}{2h}\\&\approx\frac{\frac{f(x+h,y+h)-f(x-h,y+h)}{2h}-\frac{f(x+h,y-h)-f(x-h,y-h)}{2h}}{2h}\\&=\frac{f(x+h,y+h)-f(x-h,y+h)-f(x+h,y-h)+f(x-h,y-h)}{4h^2}\end{aligned} \tag{34}$$

7 Differential equations

7.1 Ordinary differential equations

An *ordinary differential equation* (ODE) relates a single independent variable, e.g., *x*, to a function $f(x)$ and its derivatives $f'(x), f''(x),\ldots$. Most physical laws are expressed in terms of differential equations, hence their great importance. Certain classes of ODEs can be solved analytically but many cannot. In either case our derivative formulas can be used to develop numerical solutions.

Suppose a physical problem is described by a differential equation of the form

$$f''+2f'+17f=0 \tag{35}$$

One can verify that

$$f(x)=e^{-x}\cos(4x) \tag{36}$$

solves (35) by taking derivatives and substituting into the equation. A numerical approximation to (35) is given by (using (30) and (13))

$$\frac{f(x+h)-2f(x)+f(x-h)}{h^2}+2\frac{f(x+h)-f(x-h)}{2h}+17f(x)=0 \tag{37}$$

Solving this for $f(x+h)$ we obtain

$$f(x+h)=\frac{(2-17h^2)f(x)-(1-h)f(x-h)}{1+h} \tag{38}$$

Let's use this to calculate $f(x)$ for $x=0,h,2h,3h,\ldots$. To get started we need the first two values

$$f(x_1=0)=1\,,\,f(x_2=h)=e^{-h}\cos(4h) \tag{39}$$

Then we can apply (38) to get $f(x_3=x_2+h)$, $f(x_4=x_3+h)$ and so on as long as we wish. In Scilab this looks something like

```
h = 0.1;
x = 0:h:5;
f(1) = 1;
f(2) = exp(-h)*cos(4*h);
for i=2:n-1
  y(i+1) = ((2-17*h^2)*y(i)-(1-h)*y(i-1))/(1+h);
end
```

The resulting numerical solution and the exact solution are shown in Fig. 4. The agreement is excellent.

Function `odeCentDiff` in the Appendix uses this idea to numerically solve a second-order equation of the form

$$y''+p(x)y'+q(x)y=r(x)$$

Given and initial *x* value x_1, a step size *h* and the two function values $y(x_1)$ and $y(x_1+h)$. Fig. 4 compares the numerical solution of (35) using `odeCentDiff` with the exact solution

$$y=f(x)=e^{-x}\cos(4x) \tag{40}$$

for a step size $h=0.1$.

7.2 Partial differential equations

A *partial differential equation* (PDE) relates two or more independent variables, e.g., *x,y*, to a function $f(x,y)$ and its partial derivatives $\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial^2 f}{\partial x^2},\frac{\partial^2 f}{\partial x\,\partial y},\ldots$. One of the most important PDEs is *Laplace's equation*

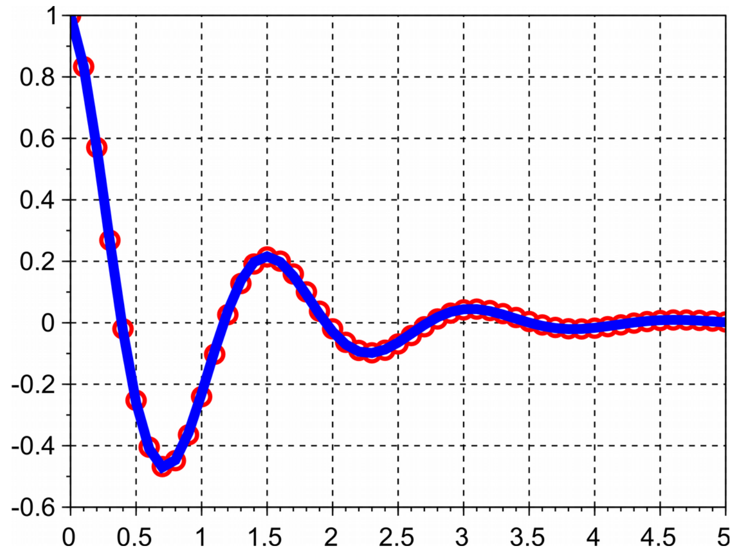

Fig. 4: Solid line: numerical solution of (35); circles: exact solution.

$$\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}=0 \tag{41}$$

Numerically we can write

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2} \\ &\approx \frac{f(x+h,y)-2f(x,y)+f(x-h,y)}{h^2}+\frac{f(x,y+h)-2f(x,y)+f(x,y-h)}{h^2} \\ &=\frac{f(x+h,y)+f(x-h,y)+f(x,y+h)+f(x,y-h)-4f(x,y)}{h^2}\end{aligned} \tag{42}$$

The last expression is zero when

$$f(x,y)=\frac{1}{4}[f(x+h,y)+f(x-h,y)+f(x,y+h)+f(x,y-h)] \tag{43}$$

which relates the value $f(x,y)$ to its “neighboring” values. Specifically $f(x,y)$ is equal to the average of its neighbor's values.

8 Appendix – Scilab code

8.1 2nd order central difference

```
//////////////////////////////////////////////////////////////////////
// derivSecondOrder.sci
// 2014-11-15, Scott Hudson, for pedagogic purposes
// Numerical estimation of derivative of f(x) using 2nd-order
// accurate central difference and "optimum" step size.
//////////////////////////////////////////////////////////////////////
function yp=derivSecondOrder(f, x)
  h = 1e-5*(1+abs(x)); //step size scales with x, no less than 1e-5
  yp = (f(x+h)-f(x-h))/(2*h);
endfunction
```

8.2 4th order central difference

```
//////////////////////////////////////////////////////////////////////
// derivFourthOrder.sci
// 2014-11-15, Scott Hudson, for pedagogic purposes
// Numerical estimation of derivative of f(x) using 4th-order
// accurate central difference and "optimum" step size.
//////////////////////////////////////////////////////////////////////
function yp=derivFourthOrder(f, x)
  h = 1e-3*(1+abs(x)); //step size scales with x, no less than 1e-3
  yp = (f(x-2*h)-8*f(x-h)+8*f(x+h)-f(x+2*h))/(12*h);
endfunction
```

8.3 Differential equation solver using 2nd order central difference

```
//////////////////////////////////////////////////////////////////////
// odeCentDiff.sci
// 2014-11-15, Scott Hudson, for pedagogic purposes
// Uses 2nd-order accurate central difference approximation to
// derivatives to solve ode y''+p(x)y'+q(x)y=r(x)
// approximations are
// y' = (y(x+h)-y(x-h))/(2h) and y'' = (y(x+h)-2y(x)+y(x-h))/h^2
// p,q,r are functions, x1 is the initial x value, h is step size,
// n is number of points to solve for, y1=y(x1), y2=y(x1+h).
//////////////////////////////////////////////////////////////////////
function [x, y]=odeCentDiff(p, q, r, x1, h, n, y1, y2)
  x = zeros(n,1);
  y = zeros(n,1);
  x(1) = x1;
  x(2) = x(1)+h;
  y(1) = y1;
  y(2) = y2;
  h2 = h*h;
  for i=2:n-1
    hp = h*p(x(i));
    x(i+1) = x(i)+h;
    y(i+1) = (2*h2*r(x(i))+(4-2*h2*q(x(i)))*y(i)+(hp-2)*y(i-1))/(2+hp);
  end
endfunction
```