1. Evaluate the integral∫10x√1+x2dxusing the Trapezoid and Simpson’s srules with 4 subdivisions of the integration limits. Later use the Gaussquadrature rule with 4 integration points. Compare the results.

%% Trapezoid

nt = 4;

at = linspace (xmin, xmax, nt+1);

yt = subs (eqs, x, at);

ht = at(2) - at(1);

valor\_numericot = ht \* (yt(1)/2 + sum(yt(2:nt)) + yt(nt+1)/2);

vpa(valor\_numericot, 4)



%% Simpson's

ns = 4;

as = linspace (xmin, xmax, ns\*2+1);

asp = as(linspace(2, ns\*2, ns));

asi = as(linspace(1, ns\*2-1, ns));

ys = subs (eqs, x, as);

ysi= subs (eqs, x, asi);

ysp= subs (eqs, x, asp);

hs = as(2) - as(1);

valor\_numericos = hs/3 \* (ys(1) + 4\*sum(ysp) + 2\*sum(ysi(2:ns)) + ys(ns\*2+1));

vpa(valor\_numericos, 4)



%% Gauss

quadrature\_4 = [

-0.3399810435848562648026658 0.6521451548625461426269361

0.3399810435848562648026658 0.6521451548625461426269361

-0.8611363115940525752239465 0.3478548451374538573730639

0.8611363115940525752239465 0.3478548451374538573730639];

valor\_numericog = 0.0;

for i = 1:4

f = @(t) ((xmax - xmin)\*t/2 + (xmax - xmin)/2);

eqs = @(x) x/(sqrt(1+x^2));

t = quadrature\_4(i, 1);

w = quadrature\_4(i, 2);

y = eqs(f(t));

valor\_numericog = w \* y \* (xmax - xmin)/2 + valor\_numericog;

end

vpa(valor\_numericog, 4)



2. The nodal coordinates of a 3-node bar element are(x1,y1) = (−1,−1),(x2,y2) = (1,1)and(x3,y3) = (0,1). Determine the element length analyti-cally; then use the Gauss quadrature testing different numbers of integration points. How many integration points are required to find the element length accurately?

syms xi

C = [-1 -1; 0 1; 1 1];

J = C'\* lin\_deriv (xi);

eqs = norm(J);

L = int(eqs, -1, 1);

valor\_analitico = vpa(L, 4);



quadrature\_1\_pts = [0.0 2.0];

quadrature\_2\_pts = [

-0.5773502692 1.0

0.5773502692 1.0];

quadrature\_4\_pts = [

-0.8611363116 0.3478548451

-0.3399810436 0.6521451549

0.3399810436 0.6521451549

0.8611363116 0.3478548451];

quadrature\_8\_pts = [

-0.9602898565 0.1012285363

-0.7966664774 0.2223810345

-0.5255324099 0.3137066459

-0.1834346425 0.3626837834

0.1834346425 0.3626837834

0.5255324099 0.3137066459

0.7966664774 0.2223810345

0.9602898565 0.1012285363];

N = quadrature\_8\_pts;

L = 0.00;

[s, w] = quadrature(N);

for i = 1:size(N, 1)

L = (subs(eqs, xi, s(i))) \* w(i) + L;

end

valor\_numerico = vpa(L, 4);



function dn = lin\_deriv (xi)

dn = [xi - 1/2

xi + 1/2

-2\*xi];

end

function [xi, w] = quadrature (quadrature)

Npst = size(quadrature, 1);

for i = 1:Npst

xi(i,1) = quadrature (i, 1);

w(i,1) = quadrature (i, 2);

end

end

3. The nodal coordinates of a 4-node quadrilateral element are(x1,y1) =(1,1),(x2,y2) = (5,2),(x3,y3) = (6,5)and(x4,y4) = (2,4). Determine the element area using the Gauss quadrature with 1, 4, and 9 integration points.Compare the results.

syms xi eta

C = [1 1; 5 2; 6 5; 2 4];

dn = quad4\_deriv (xi, eta);

J = C' \* dn;

eqs = det(J);

A = int(int(eqs, xi, -1, 1), eta, -1, 1);

valor\_analitico = vpa(A);



quadrature\_1\_pts = [0.0 0.0 4.0];

quadrature\_4\_pts = [

-0.577350269189626 -0.577350269189626 1.0

0.577350269189626 -0.577350269189626 1.0

-0.577350269189626 0.577350269189626 1.0

0.577350269189626 0.577350269189626 1.0];

quadrature\_9\_pts = [

-0.774596669241483 -0.774596669241483 0.3086419753086419

0.0 -0.774596669241483 0.4938271604938271

0.774596669241483 -0.774596669241483 0.3086419753086419

-0.774596669241483 0.0 0.4938271604938271

0.0 0.0 0.7901234567901234

0.774596669241483 0.0 0.4938271604938271

-0.774596669241483 0.774596669241483 0.3086419753086419

0.0 0.774596669241483 0.4938271604938271

0.774596669241483 0.774596669241483 0.3086419753086419];

N = (quadrature\_9\_pts);

L = 0.00;

[s, r, w] = quadrature(N);

for i = 1:size(N, 1)

L = (subs(eqs, [xi eta], [s(i) r(i)])) \* w(i) + L;

end

valor\_numerico = vpa(L, 4);



function dn = quad4\_deriv (xi, eta)

n = [1.0/4.0 \* (1 - xi) \* (1 - eta)

1.0/4.0 \* (1 + xi) \* (1 - eta)

1.0/4.0 \* (1 + xi) \* (1 + eta)

1.0/4.0 \* (1 - xi) \* (1 + eta)];

dn = [diff(n, xi), diff(n, eta)];

end

function [xi, eta, w] = quadrature (quadrature)

Npst = size(quadrature, 1);

for i = 1:Npst

xi(i, 1) = quadrature (i, 1);

eta(i, 1) = quadrature (i, 2);

w(i, 1) = quadrature (i, 3);

end

end

4. Compute the surface area shown at right. How many integration pointsare required to find the area with precision of 0.01?

syms xi eta

C = [0 0 1; 1 0 1; 1 1 0; 0 1 1];

dn = quad4\_deriv (xi, eta);

J = C' \* dn;

NormJ = sqrt((det(J([1 2], [1 2])))^2 + ...

(det(J([2 3], [1 2])))^2 + ...

(det(J([3 1], [1 2])))^2);

A = int(int(NormJ, xi, -1, 1), eta, -1, 1);

valor\_analitico = vpa(A);



quadrature\_1\_pts = [0.0 0.0 4.0];

quadrature\_4\_pts = [

-0.577350269189626 -0.577350269189626 1.0

0.577350269189626 -0.577350269189626 1.0

-0.577350269189626 0.577350269189626 1.0

0.577350269189626 0.577350269189626 1.0];

quadrature\_9\_pts = [

-0.774596669241483 -0.774596669241483 0.3086419753086419

0.0 -0.774596669241483 0.4938271604938271

0.774596669241483 -0.774596669241483 0.3086419753086419

-0.774596669241483 0.0 0.4938271604938271

0.0 0.0 0.7901234567901234

0.774596669241483 0.0 0.4938271604938271

-0.774596669241483 0.774596669241483 0.3086419753086419

0.0 0.774596669241483 0.4938271604938271

0.774596669241483 0.774596669241483 0.3086419753086419];

N = (quadrature\_4\_pts);

A = 0.00;

[s, r, w] = quadrature(N);

for i = 1:size(N, 1)

A = (subs(NormJ, [xi eta], [s(i) r(i)])) \* w(i) + A;

end

valor\_numerico = vpa(A);



function dn = quad4\_deriv (xi, eta)

n = [1.0/4.0 \* (1 - xi) \* (1 - eta)

1.0/4.0 \* (1 + xi) \* (1 - eta)

1.0/4.0 \* (1 + xi) \* (1 + eta)

1.0/4.0 \* (1 - xi) \* (1 + eta)];

dn = [diff(n, xi), diff(n, eta)];

end

function [xi, eta, w] = quadrature (quadrature)

Npst = size(quadrature, 1);

for i = 1:Npst

xi(i, 1) = quadrature (i, 1);

eta(i, 1) = quadrature (i, 2);

w(i, 1) = quadrature (i, 3);

end

end