

Bending

U2

$2.5 \times 10^{-5}$

2.5

2

1.5

1

0.5

U2

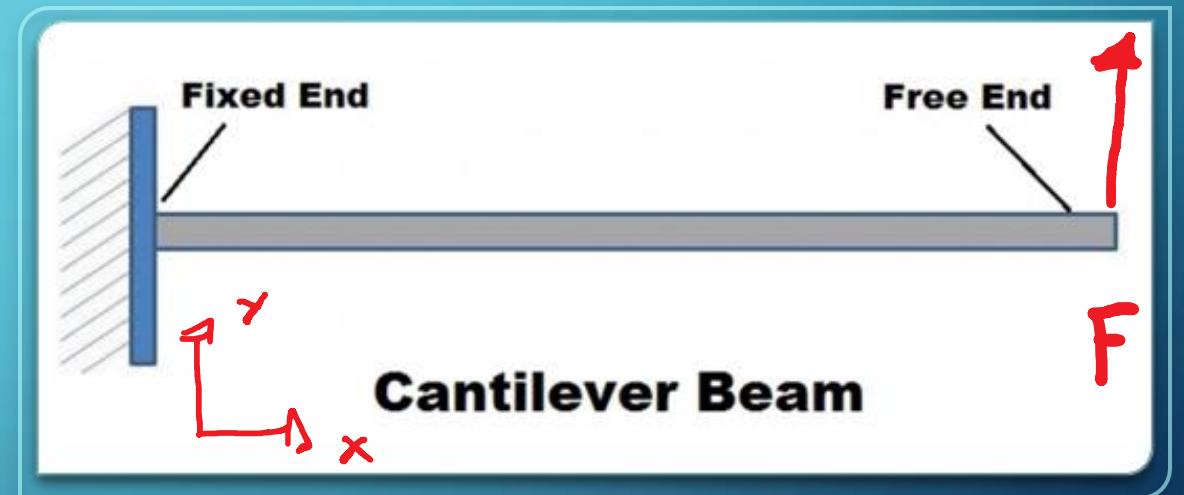
# COMPARAÇÃO DOS RESULTADOS

ABAQUS VS FEM

- FEM order 1 (Blue)
- Abaqus order 1 (Red)
- FEM order 2 (Green)
- Abaqus order 2 (Purple)

# PARÂMETROS

- Comprimento = 10 [m]
- Altura = 1 [m]
- $F = 11e2$  [N]
- Elementos em  $x = 40$
- Elementos em  $y = 10$



case 2

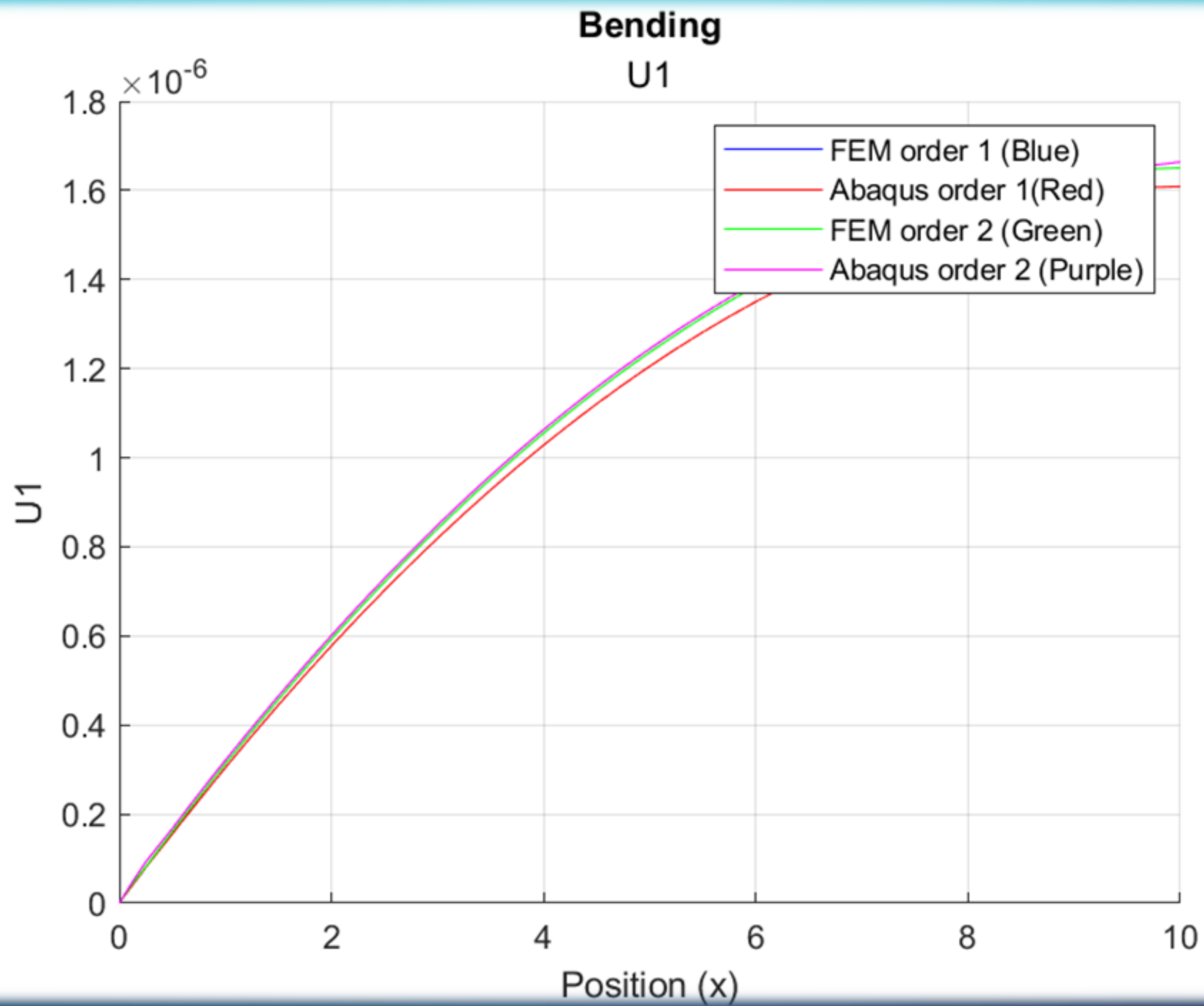
```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
        for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
            for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, icomp);
                s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, icomp);
            end
            mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy);
            med(index, 1) = med(index, 1) + 1;
            inode = inode + 1;
        end
    end
end
```

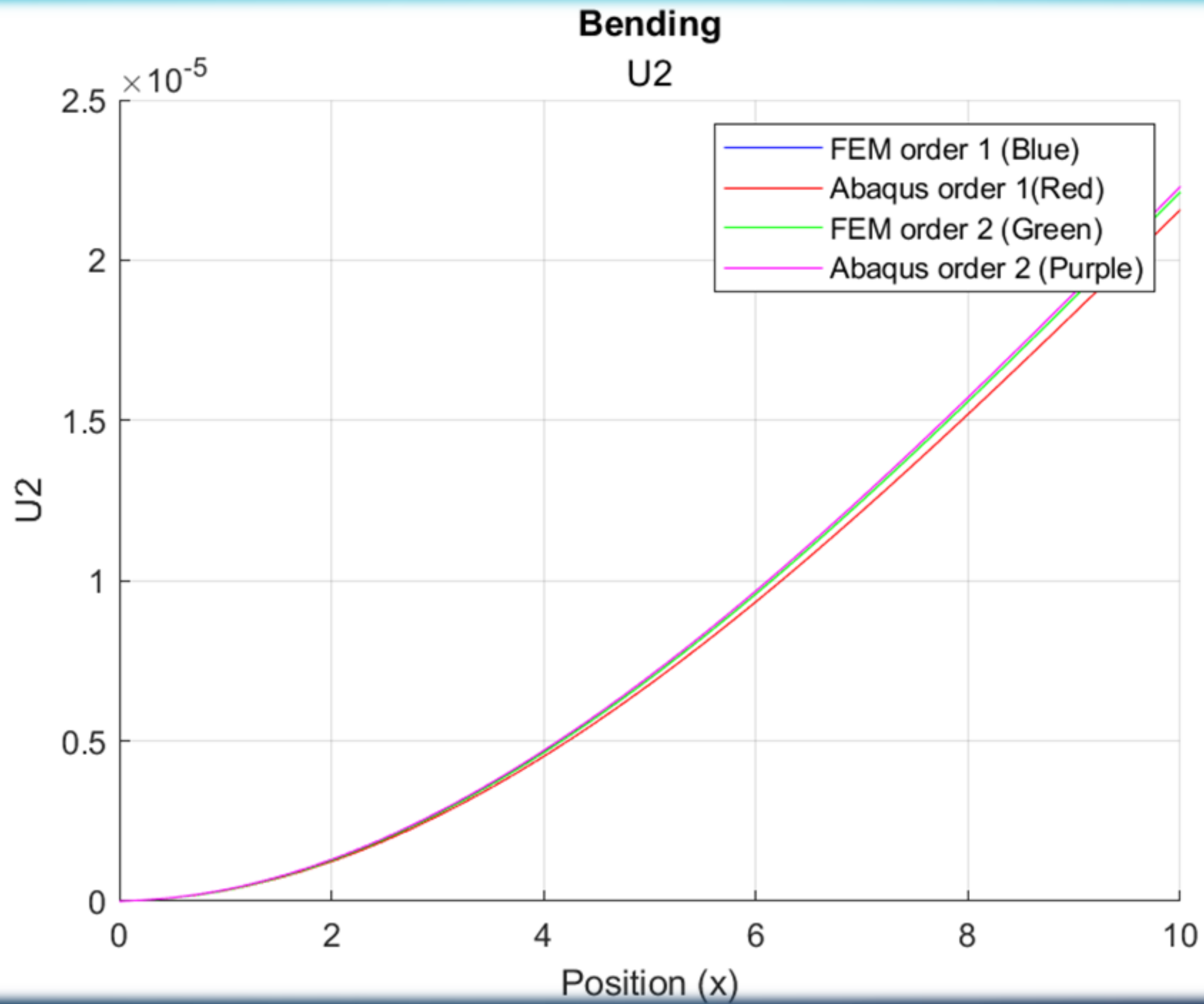
# RESULTADOS

## DESLOCAMENTO

case 3

```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwz = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in z
        for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
            for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
                for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                    index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                    e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, iwz, icomp);
                    s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, iwz, icomp);
                end
                mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy, iwz);
                med(index, 1) = med(index, 1) + 1;
                inode = inode + 1;
            end
        end
    end
end
```





case 2

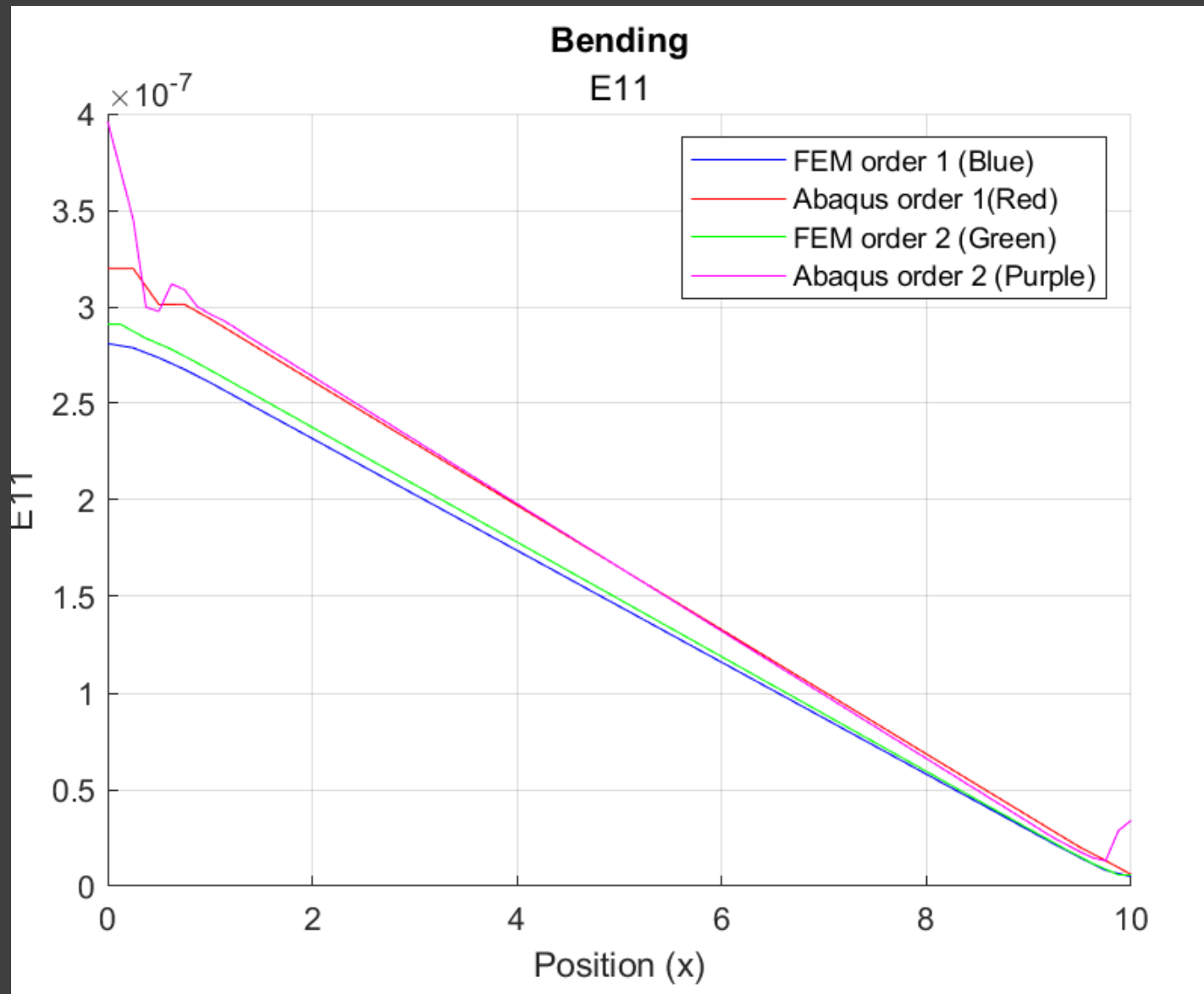
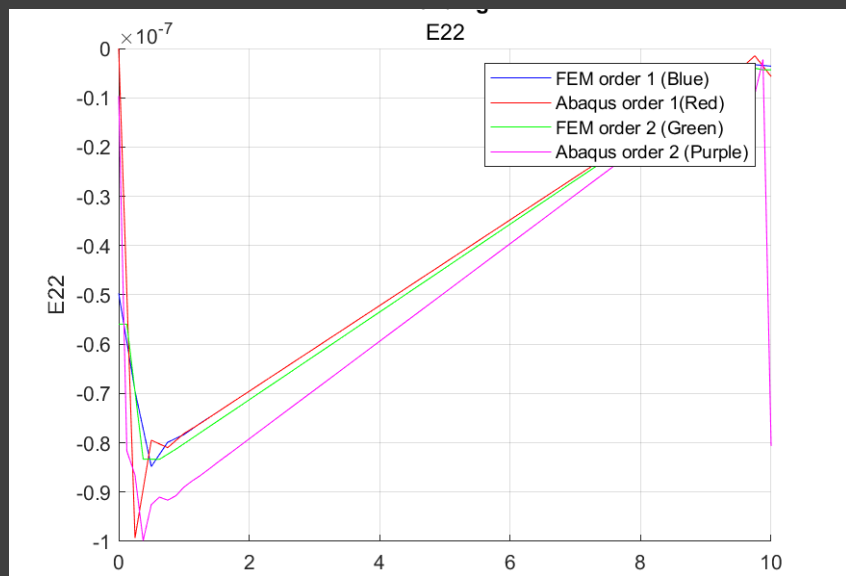
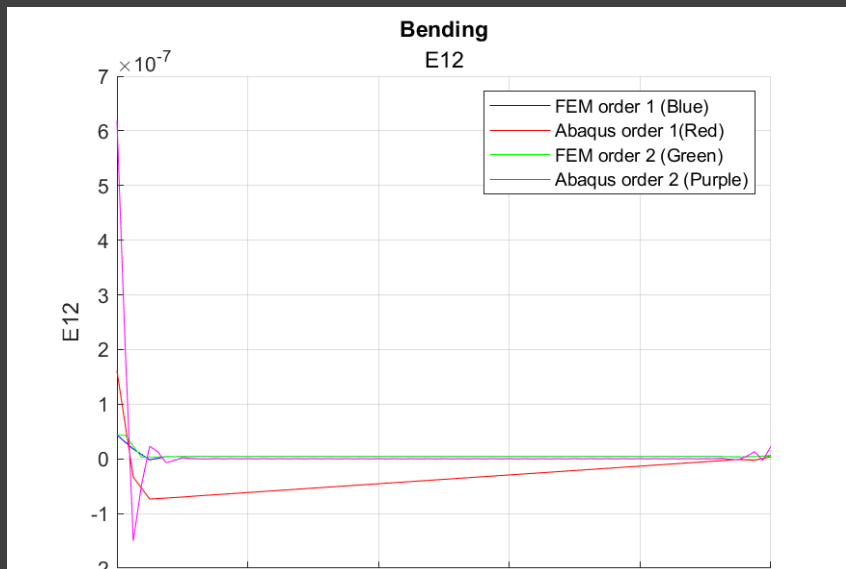
```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
        for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
            for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, icomp);
                s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, icomp);
            end
            mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy);
            med(index, 1) = med(index, 1) + 1;
            inode = inode + 1;
        end
    end
end
```

# RESULTADOS

## DEFORMAÇÃO

case 3

```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwz = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in z
        for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
            for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
                for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                    index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                    e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, iwz, icomp);
                    s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, iwz, icomp);
                end
                mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy, iwz);
                med(index, 1) = med(index, 1) + 1;
                inode = inode + 1;
            end
        end
    end
end
```



case 2

```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
        for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
            for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, icomp);
                s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, icomp);
            end
            mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy);
            med(index, 1) = med(index, 1) + 1;
            inode = inode + 1;
        end
    end
end
```

# RESULTADOS

## TENSÃO

case 3

```
for iel = 1:size(obj.Matrix.Pre.Mesh.Connectivity, 1) % for each element
    inode = 1; % for each node in the element
    for iwz = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in z
        for iwy = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in y
            for iwx = 1:obj.Matrix.Pre.Mesh.Order+1 % for each integration point in x
                for icomp = 1:(obj.Matrix.Pre.Mesh.Dim+(obj.Matrix.Pre.Mesh.Dim^2-obj.Matrix.Pre.Mesh.Dim)/2) % for each comp
                    index = sort(obj.Matrix.Pre.Mesh.Connectivity(iel, 2:end));
                    e(index, icomp) = e(index, icomp) + obj.E(iel, iwx, iwy, iwz, icomp);
                    s(index, icomp) = s(index, icomp) + obj.S(iel, iwx, iwy, iwz, icomp);
                end
                mis(index, 1) = mis(index, 1) + obj.Mises(iel, iwx, iwy, iwz);
                med(index, 1) = med(index, 1) + 1;
                inode = inode + 1;
            end
        end
    end
end
```



