

LAMPIRAN

Lampiran A Perhitungan Manual

A.1. Data Pemboran

Surface Section

Casing OD =	20	in
Casing length =	750	ft
Total Joint	22	
Fracture gradient =	13.68	ppg
Mud density while setting casing =	8.5	ppg
Formation pressure =	370.5	psi
Temperature =	17.61	oF
Gas deviation factor	0.93	
Design factor =	Burst, Collapse	1.1
	Tension	1.6

Intermediate Section

Casing OD =	13 3/8	in
Casing length =	3172.11	ft
Total Joint	91	
Fracture gradient =	14.8	ppg
Heaviest mud density =	12.5	ppg
Mud density while setting casing =	10.5	ppg
Formation pressure =	1567.22	psi
Temperature =	74.46	oF
Gas deviation factor	0.93	
Design factor =	Burst, Collapse	1.1 (assume)
	Tension	1.6 (assume)
L=	2269.2	ft

Production Section

Casing OD =	9 5/8	in
Casing length =	7092.98	ft
Total Joint	203	
Fracture gradient =	15.03	ppg
Heaviest mud density =	14.5	ppg
Mud density while setting casing =	13.7	ppg
Formation pressure =	3503.9	psi
Temperature =	166.512	oF
Gas deviation factor	0.93	
Design factor =	Burst, Collapse	1.1 (assume)
	Tension	1.6 (assume)
L=	4374.17	ft

A.2 Perhitungan

Surface Section

Kedalaman (ft)	Beban burst	Burst resistance	Beban	tension resistance	beban collapse	collapse resistance	biaxial
0	534.897	1530	98161.8321	1077000	0	520	494.68
750	246.147	1530	0	1077000	364.65	520	494.68

Intermediate Section

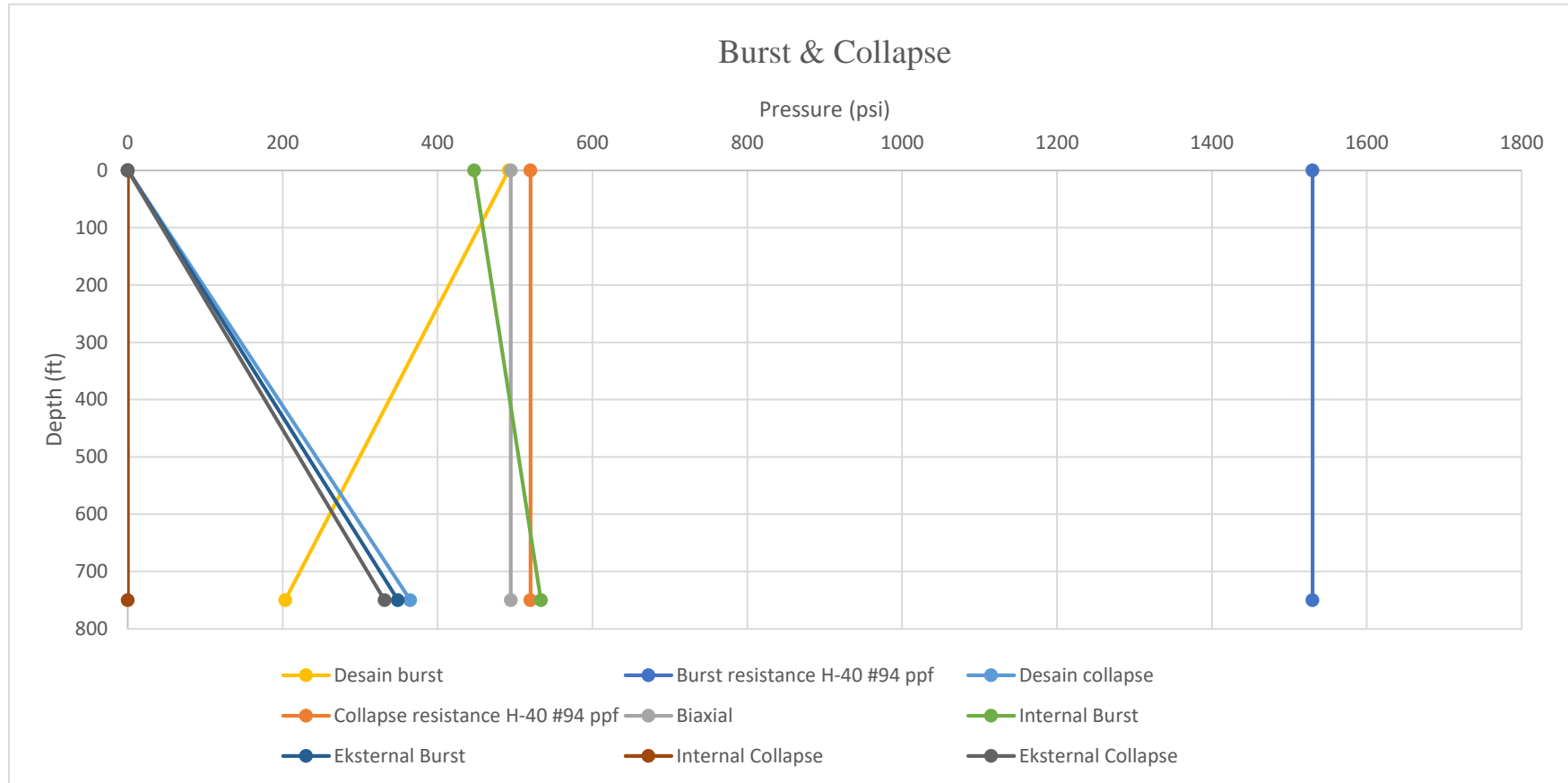
kedalaman (ft)	beban burst (psi)	burst resistance	beban tension	tension resistance	beban collapse	collapse resistance	biaxial
0	2465.5542	2730	232266.2528	853000	0	1130	944.286
902.91		2730			542.287746	1130	944.286
3172	1244.2919	2730	0	853000	282.691266	1130	944.286

Production Section

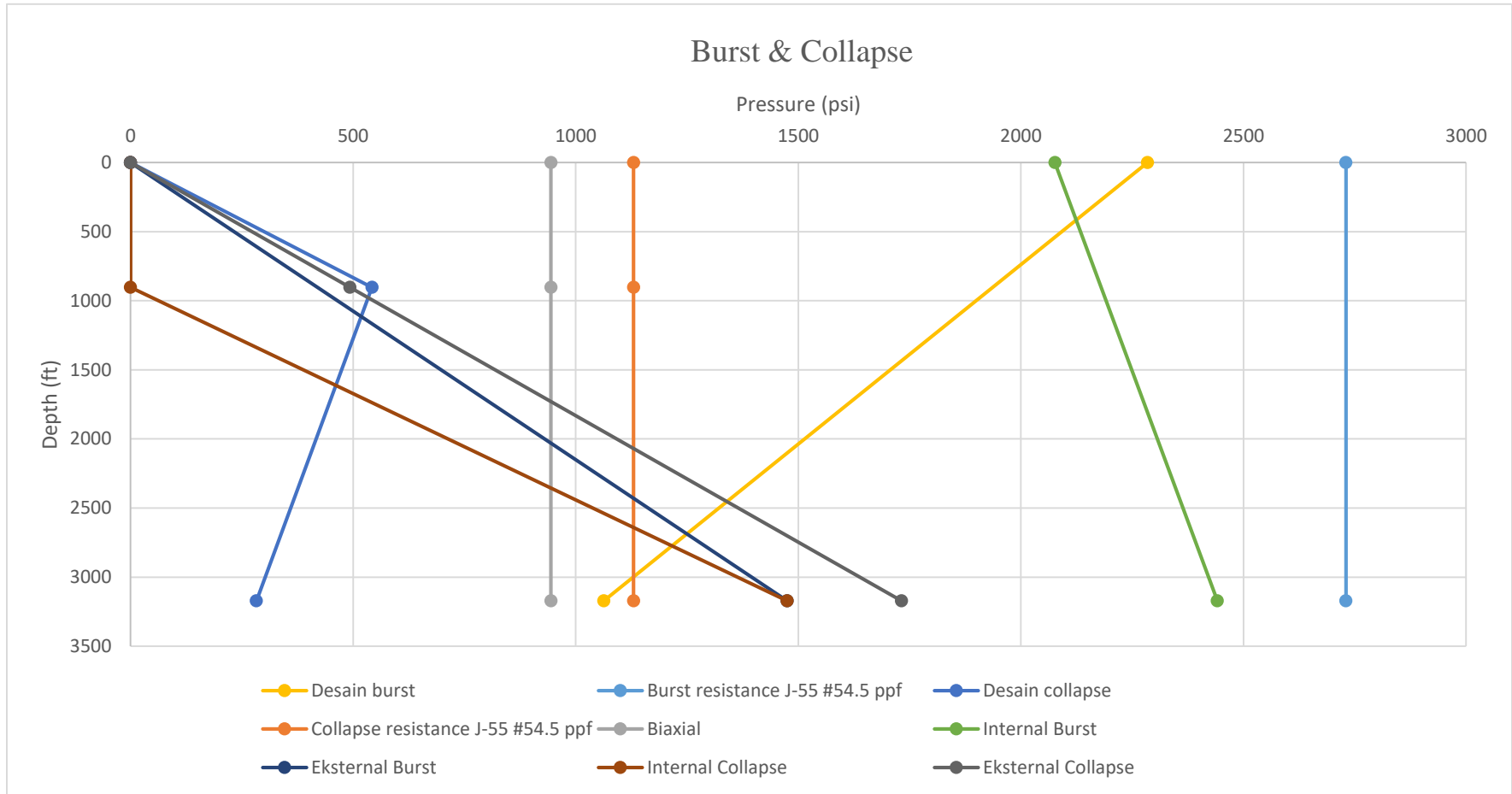
kedalaman (ft)	beban burst	burst resistance	beban tension	tension resistance	beban collapse	collapse resistance	biaxial
0	5606.40488	5750	359002.2488	916000	0	3090	2301.04
2718.81261					2130.570313	3090	2301.04
7029.98	5629.811714	5750	0	916000	1930.408414	3090	2301.04

A.3 Grafik

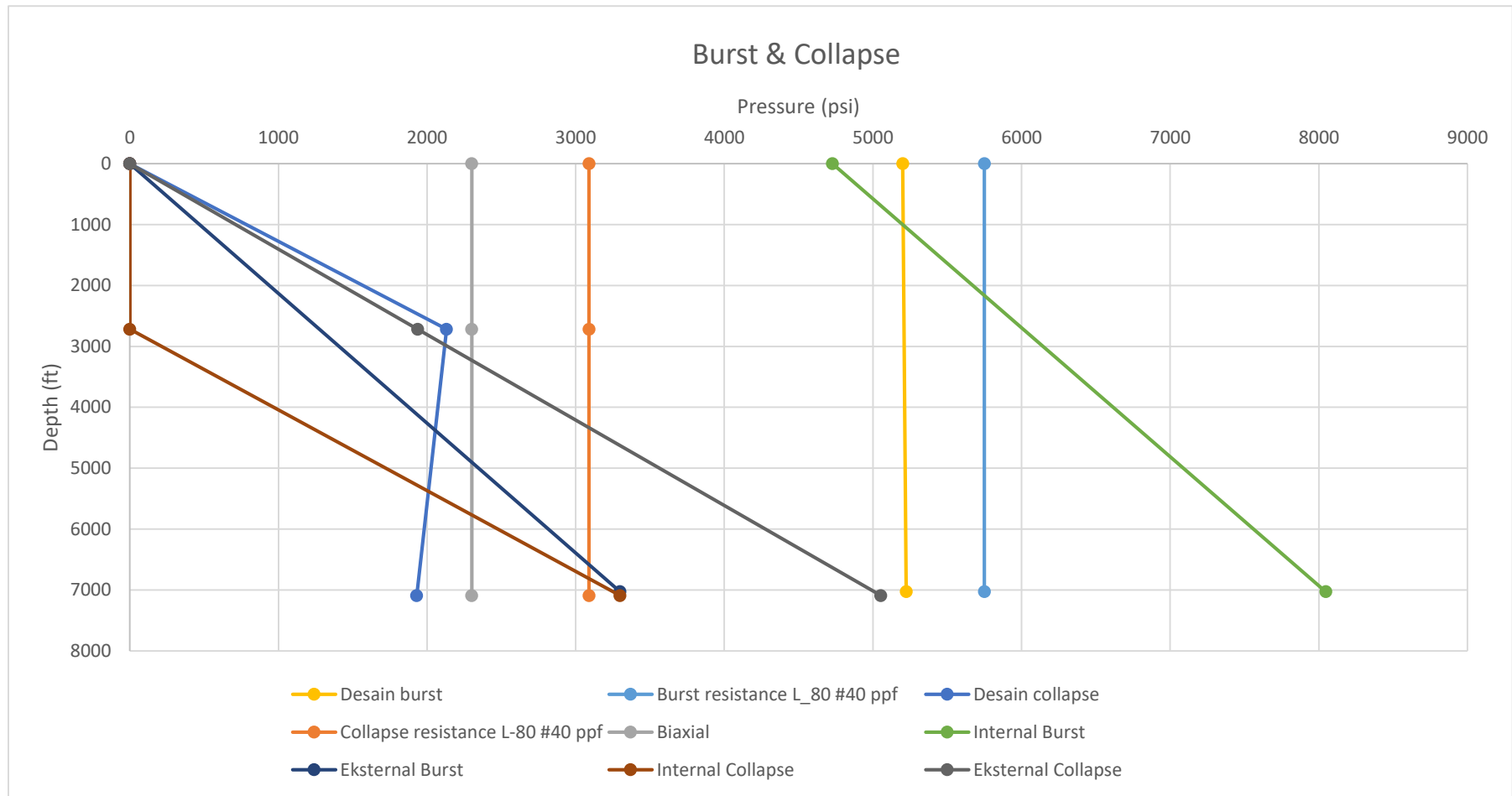
Surface



Intermediate



Production

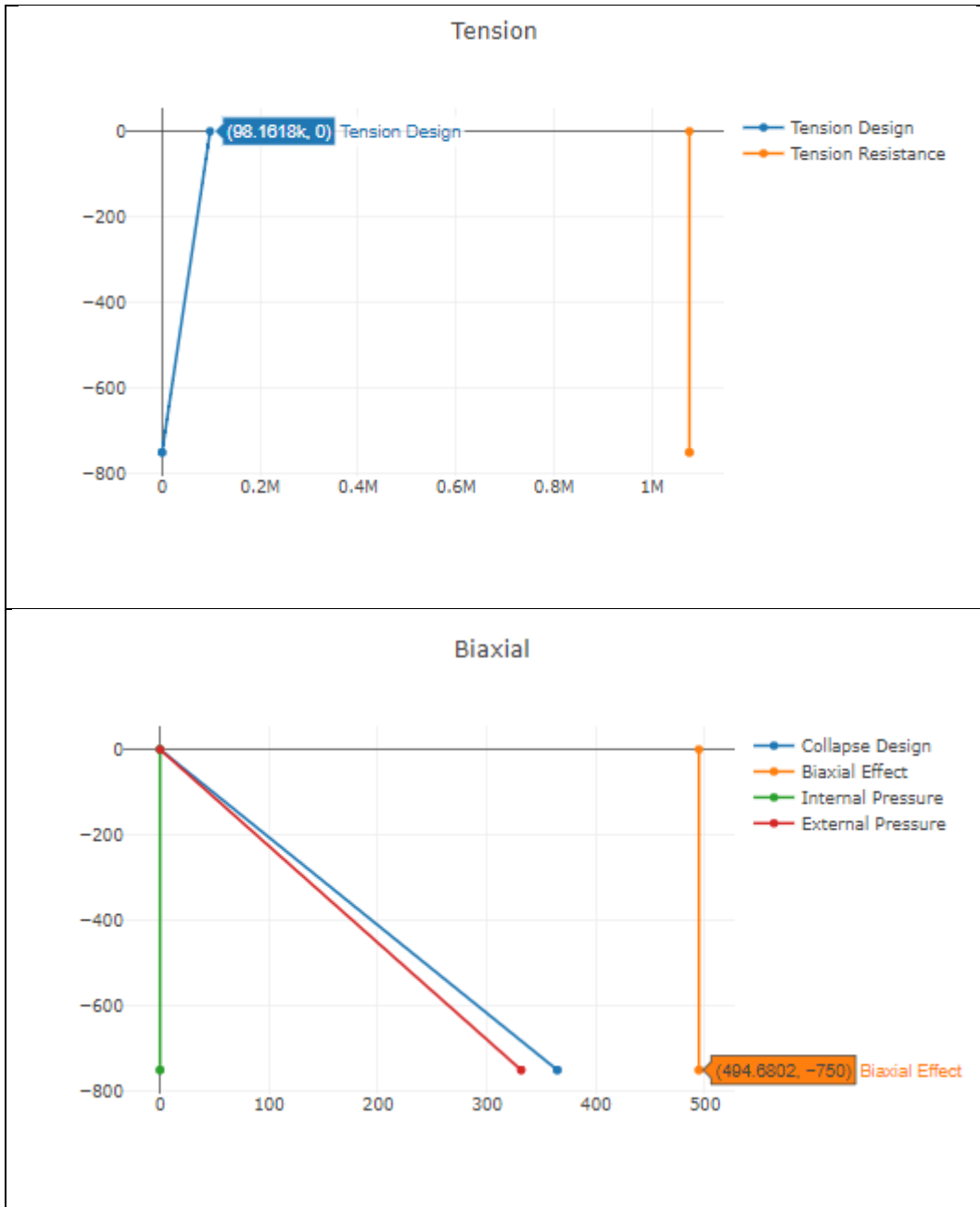


Lampiran B

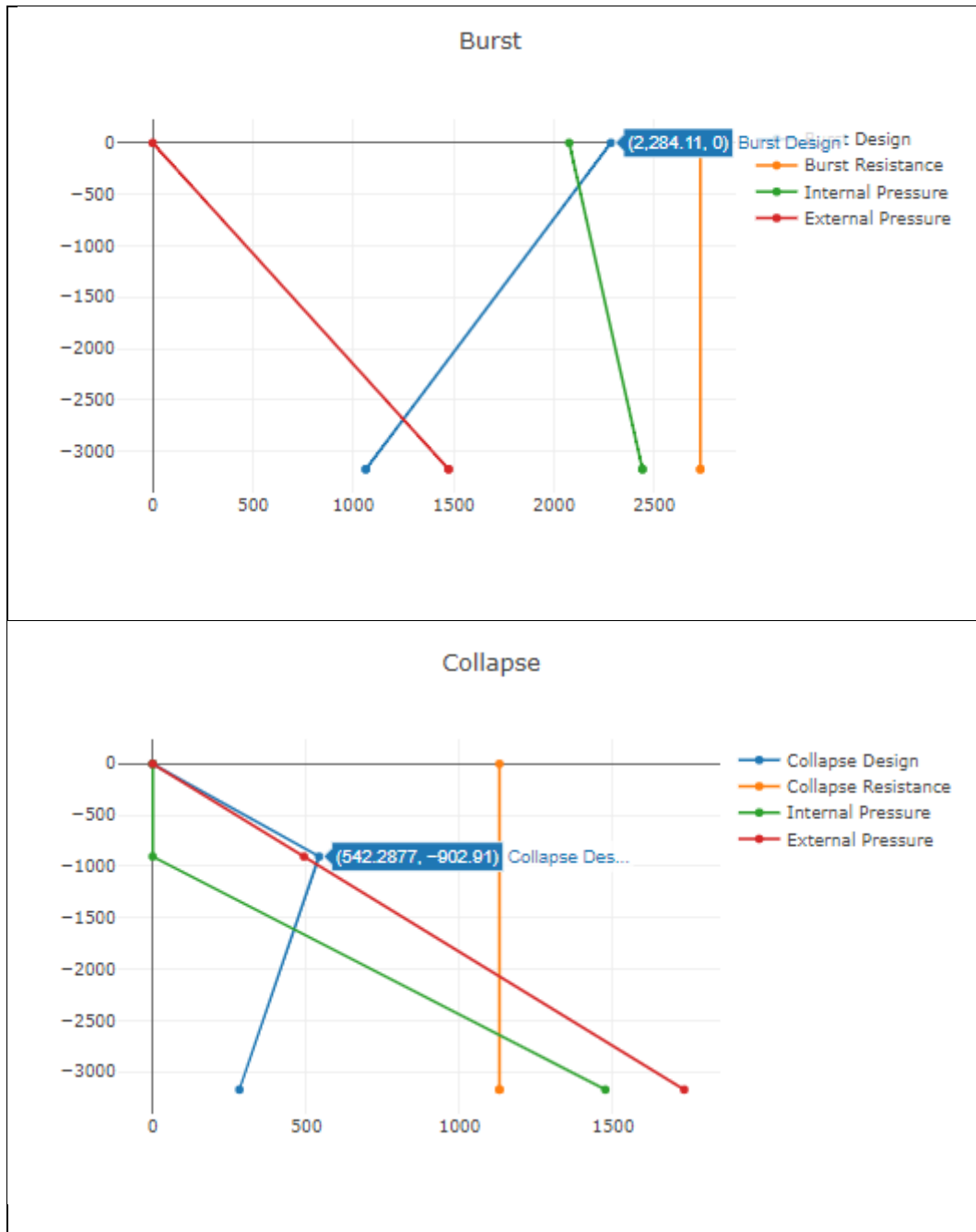
Perhitungan Program

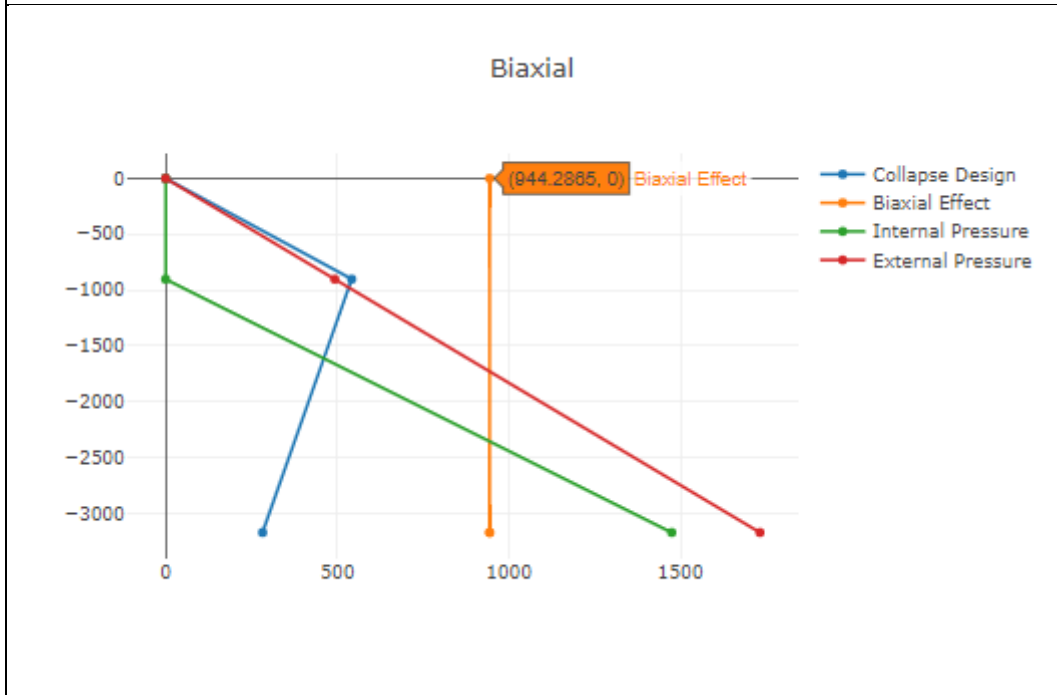
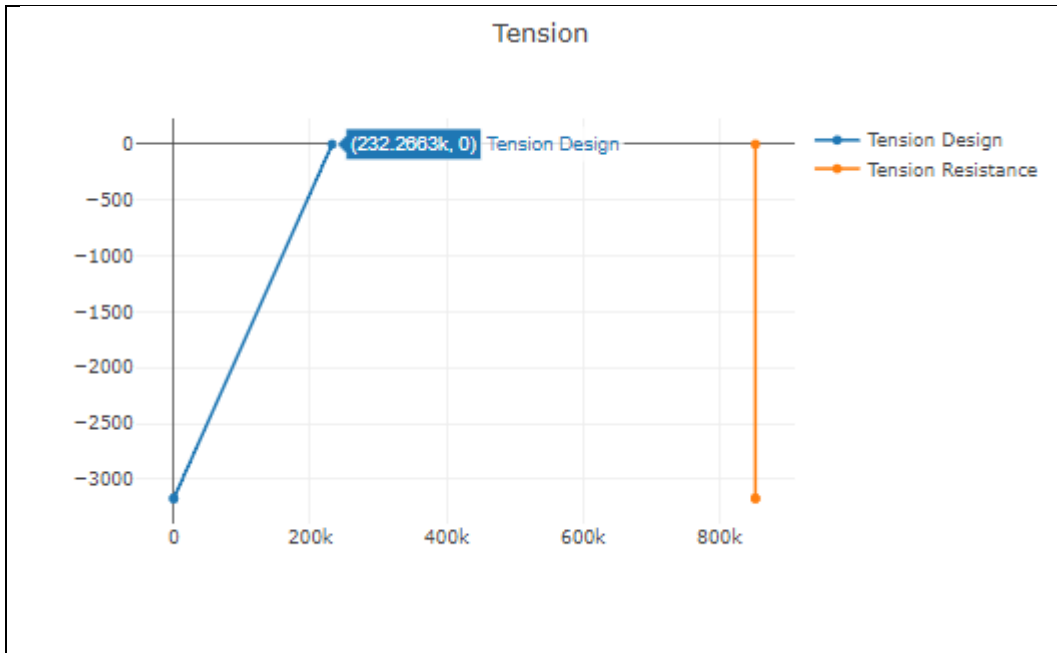
Surface



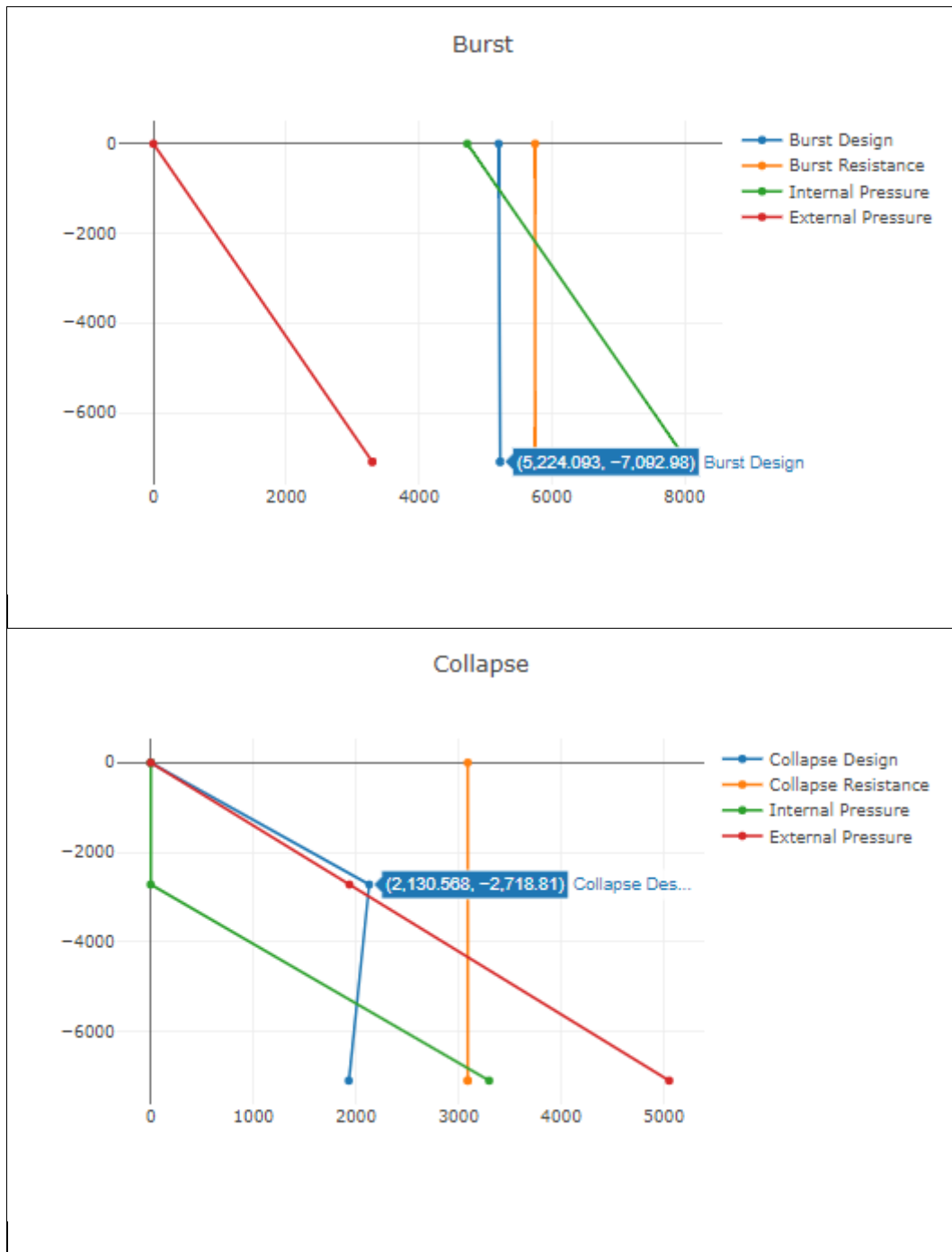


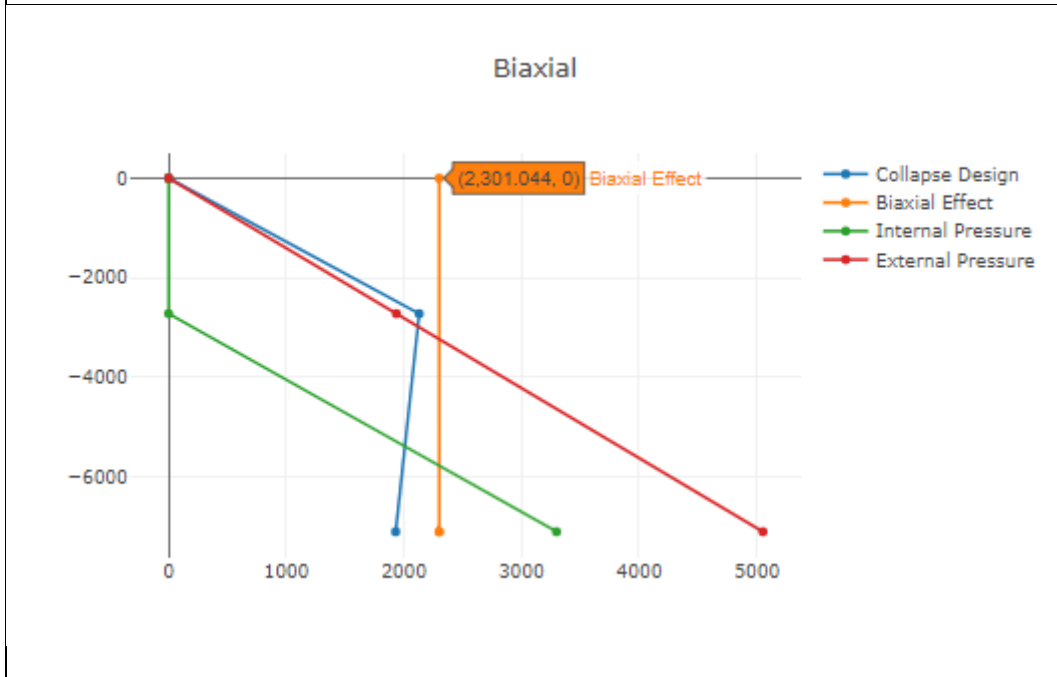
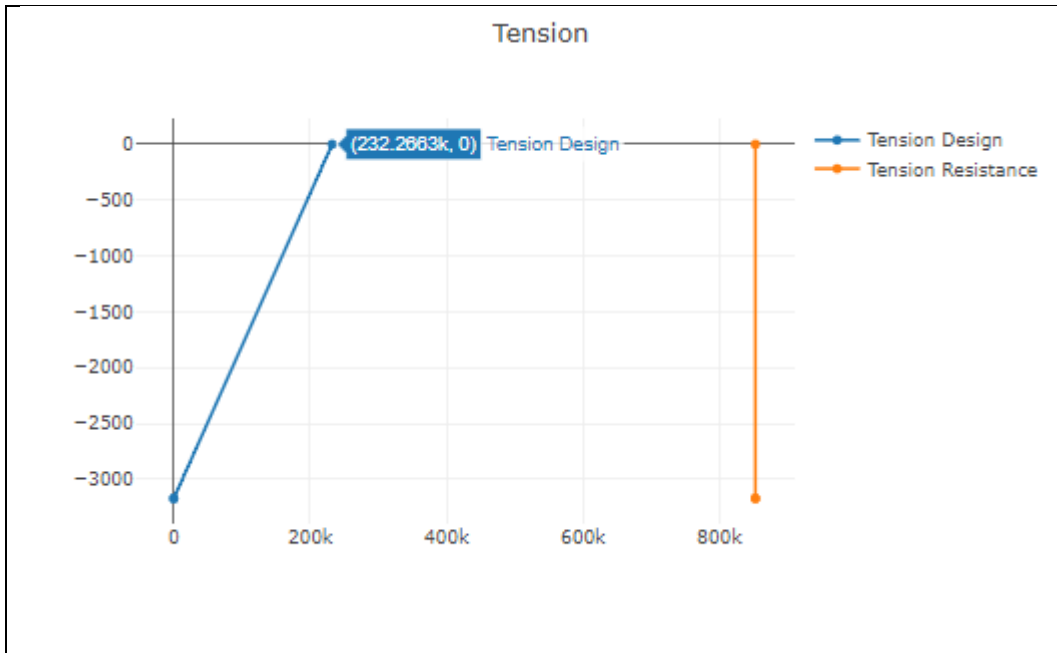
Intermediate





Production





Lampiran C

Coding Program

C.1 Persamaan yang Digunakan

```
export const useCalculation = ({ data, currentTab, customData }:
Props) => {
  // SELECT
  const odSelect = useData({ type: "select", key: "od", data,
currentTab })
  const wnSelect = useData({ type: "select", key: "wn", data,
currentTab })
  const gradeSelect = useData({
    type: "select",
    key: "grade",
    data,
    currentTab,
  })

  const dataByGrade = newConstant?.find(
    (v) =>
      v?.OD === odSelect &&
      v?.NOMINAL_WEIGHT === wnSelect &&
      v?.GRADE === gradeSelect
  )!

  const od = +(customData?.od ?? odSelect)
  const wn = +(customData?.wn ?? wnSelect)
  const ym = customData?.ym ?? +dataByGrade?.BODY_YIELD
  const pe = customData?.pe ?? +dataByGrade?.IYM_PE
  const t = customData?.t ?? +dataByGrade?.WALL
  const id = customData?.id ?? +dataByGrade?.ID

  // INPUT
  const pf = +useData({ type: "input", key: "pf", data, currentTab
})
  const td = +useData({ type: "input", key: "td", data, currentTab
})
  const csd = +useData({ type: "input", key: "csd", data, currentTab
})
  const l = +useData({ type: "input", key: "l", data, currentTab })
  const pp = +useData({
    type: "input",
    key: "pp",
    data,
    currentTab: "production",
```

```

    })
    const mudDensity = +useData({
      type: "input",
      key: "mud-density",
      data,
      currentTab,
    })
    const nextMudDensity = +useData({
      type: "input",
      key: "next-mud-density",
      data,
      currentTab,
    })
    const overpull = +useData({
      type: "input",
      key: "overpull",
      data,
      currentTab,
    })
    const sfBurst = +useData({ type: "input", key: "sf-burst", data,
currentTab })
    const sfCollapse = +useData({
      type: "input",
      key: "sf-collapse",
      data,
      currentTab,
    })
    const sfTension = +useData({
      type: "input",
      key: "sf-tension",
      data,
      currentTab,
    })
  })

  // CONVERTED
  const odConverted = useConvertedData({ data: od, type: "inch-to-
feet" })
  const tConverted = useConvertedData({ data: t, type: "inch-to-
feet" })
  const idConverted = useConvertedData({ data: id, type: "inch-to-
feet" })
  const ymConverted = useConvertedData({ data: ym, type: "convert-
to-kilo" })

  const exactDataFinder: TData[] = newConstant?.filter(
    (v) =>
      v?.OD === od &&

```

```

    v?.NOMINAL_WEIGHT === wn &&
    v?.BODY_YIELD === ym &&
    v?.IYM_PE === pe &&
    v?.WALL === t &&
    v?.ID === id
  )

  const firstBurstSurfaceAndIntermediate = (() => {
    let value = 0
    if (currentTab === "surface" || currentTab === "intermediate")
      value = (0.052 * (pf + 1) * csd - (0.115 * td)) * sfBurst
    return value
  })()

  const secondBurstSurfaceAndIntermediate = (() => {
    let value = 0
    if (currentTab === "surface" || currentTab === "intermediate")
      value = (0.052 * (pf + 1) * csd - 0.115 * (td - csd) - 0.465 *
csd) * sfBurst
    return value
  })()

  const firstBurstProduction = (() => {
    let value = 0
    if (currentTab === "production")
      value = (0.052 * (pf + 1) * csd - 0.115 * csd) * sfBurst
    return value
  })()

  const firstBurstProductionWithoutSFBurst = (() => {
    let value = 0
    if (currentTab === "production")
      value = 0.052 * (pf + 1) * csd - 0.115 * csd
    return value
  })()

  const secondBurstProduction = (() => {
    let value = 0
    if (currentTab === "production")
      value =
        (firstBurstProductionWithoutSFBurst + 0.052 * pp * csd -
0.465 * csd) *
        sfBurst
    return value
  })()

  const burstResistance = (() => {

```

```

    let value = 0
    value = pe
    return value
  })()

  const firstCollapse = 0

const csdMinLCollapse = (() => {
  let value = 0
  value = (0.052 * (csd - 1) * mudDensity ) * sfCollapse
  return value
})()
const secondCollapse = (() => {
  let value = 0
  value = (0.052 * csd * mudDensity - 0.052 * 1 * nextMudDensity)
* sfCollapse
  return value
})()

const bouyantWeight = (() => {
  let value = 0
  value = wn * csd * (1 - mudDensity / 65.5) + overpull
  return value
})()

const tensionResistance = (() => {
  let value = 0
  value = ymConverted
  return value
})()

const axialLoad = (() => {
  let value = 0
  value = bouyantWeight * sfTension
  return value
})()

const biaxialEffect = (() => {
  let value = 0
  const collapsePressure =
Number(exactDataFinder?.at(0)?.COLLAPSE_PRESSURE)

  value =
    collapsePressure *
    (Math.pow(1 - (3 / 4) * Math.pow(axialLoad / ymConverted, 2),
0.5) -

```



```

        0.5 * (axialLoad / ymConverted))
    return value
  })()

  return {
    // SELECT
    od,
    wn,
    ym,
    t,
    id,
    // INPUT
    ymConverted,
    pe,
    pf,
    td,
    csd,
    pp,
    mudDensity,
    nextMudDensity,
    l,
    odConverted,
    idConverted,
    tConverted,
    exactDataFinder,
    firstBurstSurfaceAndIntermediate,
    secondBurstSurfaceAndIntermediate,
    firstBurstProduction,
    secondBurstProduction,
    burstResistance,
    firstCollapse,
    csdMinLCollapse,
    secondCollapse,
    bouyantWeight,
    tensionResistance,
    axialLoad,
    biaxialEffect,
  }
}

```

C.2 Grafik

```

export const useDataGraph = ({ data, currentTab, customData }:
Props) => {
  const {
    csd,
    l,

```

```

    exactDataFinder,
    firstBurstSurfaceAndIntermediate,
    secondBurstSurfaceAndIntermediate,
    firstBurstProduction,
    secondBurstProduction,
    burstResistance,
    firstCollapse,
    csdMinLCollapse,
    secondCollapse,
    tensionResistance,
    axialLoad,
    biaxialEffect,
  } = useCalculation({ data, currentTab, customData })

  const burstProductionDot = (() => ({
    x: [firstBurstProduction, secondBurstProduction,
secondBurstProduction],
    y: [0, -csd, -csd],
  })))()
  const burstSurfaceAndIntermediateDot = (() => ({
    x: [firstBurstSurfaceAndIntermediate,
secondBurstSurfaceAndIntermediate,
secondBurstSurfaceAndIntermediate],
    y: [0, -csd, -csd],
  })))()
  const burstResistanceDot = (() => ({
    x: [burstResistance, burstResistance, burstResistance],
    y: [0, -csd, -csd],
  })))()
  const collapseDot = (() => ({
    x: [firstCollapse,csdMinLCollapse, secondCollapse],
    y: [0, -(csd-1), -csd],
  })))()
  const collapsePressureDot = (() => ({
    x: [
      Number(exactDataFinder?.at(0)?.COLLAPSE_PRESSURE),
      Number(exactDataFinder?.at(0)?.COLLAPSE_PRESSURE),
      Number(exactDataFinder?.at(0)?.COLLAPSE_PRESSURE),
    ],
    y: [0, -csd, -csd],
  })))()
  const axialLoadDot = (() => ({
    x: [axialLoad, 0, 0],
    y: [0, -csd, -csd],
  })))()
  const tensionResistanceDot = (() => ({
    x: [tensionResistance, tensionResistance, tensionResistance],

```

```

    y: [0, -csd, -csd],
  )))()
const biaxialEffectDot = (() => ({
  x: [biaxialEffect, biaxialEffect, biaxialEffect],
  y: [0, -csd, -csd],
  )))()

const burstProductionLineCollection: TLineCollection = {
  line1: {
    x1: burstProductionDot?.x?.at(0) ?? 0,
    y1: burstProductionDot?.y?.at(0) ?? 0,
    x2: burstProductionDot?.x?.at(1) ?? 0,
    y2: burstProductionDot?.y?.at(1) ?? 0,
    x3: burstProductionDot?.x?.at(2) ?? 0,
    y3: burstProductionDot?.y?.at(2) ?? 0,
  },
  line2: {
    x1: burstResistanceDot?.x?.at(0) ?? 0,
    y1: burstResistanceDot?.y?.at(0) ?? 0,
    x2: burstResistanceDot?.x?.at(1) ?? 0,
    y2: burstResistanceDot?.y?.at(1) ?? 0,
    x3: burstResistanceDot?.x?.at(2) ?? 0,
    y3: burstResistanceDot?.y?.at(2) ?? 0,
  },
}
const burstSurfaceAndIntermediateLineCollection: TLineCollection =
{
  line1: {
    x1: burstSurfaceAndIntermediateDot?.x?.at(0) ?? 0,
    y1: burstSurfaceAndIntermediateDot?.y?.at(0) ?? 0,
    x2: burstSurfaceAndIntermediateDot?.x?.at(1) ?? 0,
    y2: burstSurfaceAndIntermediateDot?.y?.at(1) ?? 0,
    x3: burstSurfaceAndIntermediateDot?.x?.at(2) ?? 0,
    y3: burstSurfaceAndIntermediateDot?.y?.at(2) ?? 0,
  },
  line2: {
    x1: burstResistanceDot?.x?.at(0) ?? 0,
    y1: burstResistanceDot?.y?.at(0) ?? 0,
    x2: burstResistanceDot?.x?.at(1) ?? 0,
    y2: burstResistanceDot?.y?.at(1) ?? 0,
    x3: burstResistanceDot?.x?.at(2) ?? 0,
    y3: burstResistanceDot?.y?.at(2) ?? 0,
  },
}
const collapseLineCollection: TLineCollection = {
  line1: {
    x1: collapseDot?.x?.at(0) ?? 0,

```

```

    y1: collapseDot?.y?.at(0) ?? 0,
    x2: collapseDot?.x?.at(1) ?? 0,
    y2: collapseDot?.y?.at(1) ?? 0,
    x3: collapseDot?.x?.at(2) ?? 0,
    y3: collapseDot?.y?.at(2) ?? 0,
  },
  line2: {
    x1: collapsePressureDot?.x?.at(0) ?? 0,
    y1: collapsePressureDot?.y?.at(0) ?? 0,
    x2: collapsePressureDot?.x?.at(1) ?? 0,
    y2: collapsePressureDot?.y?.at(1) ?? 0,
    x3: collapsePressureDot?.x?.at(2) ?? 0,
    y3: collapsePressureDot?.y?.at(2) ?? 0,
  },
}
const tensionLineCollection: TLineCollection = {
  line1: {
    x1: axialLoadDot?.x?.at(0) ?? 0,
    y1: axialLoadDot?.y?.at(0) ?? 0,
    x2: axialLoadDot?.x?.at(1) ?? 0,
    y2: axialLoadDot?.y?.at(1) ?? 0,
    x3: axialLoadDot?.x?.at(2) ?? 0,
    y3: axialLoadDot?.y?.at(2) ?? 0,
  },
  line2: {
    x1: tensionResistanceDot?.x?.at(0) ?? 0,
    y1: tensionResistanceDot?.y?.at(0) ?? 0,
    x2: tensionResistanceDot?.x?.at(1) ?? 0,
    y2: tensionResistanceDot?.y?.at(1) ?? 0,
    x3: tensionResistanceDot?.x?.at(2) ?? 0,
    y3: tensionResistanceDot?.y?.at(2) ?? 0,
  },
}
const biaxialLineCollection: TLineCollection = {
  line1: {
    x1: collapseDot?.x?.at(0) ?? 0,
    y1: collapseDot?.y?.at(0) ?? 0,
    x2: collapseDot?.x?.at(1) ?? 0,
    y2: collapseDot?.y?.at(1) ?? 0,
    x3: collapseDot?.x?.at(2) ?? 0,
    y3: collapseDot?.y?.at(2) ?? 0,
  },
  line2: {
    x1: biaxialEffectDot?.x?.at(0) ?? 0,
    y1: biaxialEffectDot?.y?.at(0) ?? 0,
    x2: biaxialEffectDot?.x?.at(1) ?? 0,
    y2: biaxialEffectDot?.y?.at(1) ?? 0,
  },
}

```

```

    x3: biaxialEffectDot?.x?.at(2) ?? 0,
    y3: biaxialEffectDot?.y?.at(2) ?? 0,
  },
}

const burstProductionIntersection = useFindIntersection(
  burstProductionLineCollection.line1,
  burstProductionLineCollection.line2
)
const burstSurfaceAndIntermediateIntersection =
useFindIntersection(
  burstSurfaceAndIntermediateLineCollection.line1,
  burstSurfaceAndIntermediateLineCollection.line2
)
const collapseIntersection = useFindIntersection(
  collapseLineCollection.line1,
  collapseLineCollection.line2
)
const tensionIntersection = useFindIntersection(
  tensionLineCollection.line1,
  tensionLineCollection.line2
)
const biaxialIntersection = useFindIntersection(
  biaxialLineCollection.line1,
  biaxialLineCollection.line2
)

const burstProductionHigherThanBoundaries = useFindPastBoundaries(
  burstProductionLineCollection.line1,
  burstProductionLineCollection.line2
)
const burstSurfaceAndIntermediateHigherThanBoundaries =
useFindPastBoundaries(
  burstSurfaceAndIntermediateLineCollection.line1,
  burstSurfaceAndIntermediateLineCollection.line2
)
const collapseHigherThanBoundaries = useFindPastBoundaries(
  collapseLineCollection.line1,
  collapseLineCollection.line2
)
const tensionHigherThanBoundaries = useFindPastBoundaries(
  tensionLineCollection.line1,
  tensionLineCollection.line2
)
const biaxialHigherThanBoundaries = useFindPastBoundaries(
  biaxialLineCollection.line1,
  biaxialLineCollection.line2
)

```

)

```
const result: TUseDataGraph = (() => {
  return {
    dataGraph: [
      {
        type: "burstProduction",
        x: burstProductionDot?.x,
        y: burstProductionDot?.y,
      },
      {
        type: "burstSurfaceAndIntermediate",
        x: burstSurfaceAndIntermediateDot?.x,
        y: burstSurfaceAndIntermediateDot?.y,
      },
      {
        type: "burstResistance",
        x: burstResistanceDot?.x,
        y: burstResistanceDot?.y,
      },
      {
        type: "collapse",
        x: collapseDot?.x,
        y: collapseDot?.y,
      },
      {
        type: "collapsePressure",
        x: collapsePressureDot?.x,
        y: collapsePressureDot?.y,
      },
      {
        type: "axialLoad",
        x: axialLoadDot?.x,
        y: axialLoadDot?.y,
      },
      {
        type: "tensionResistance",
        x: tensionResistanceDot?.x,
        y: tensionResistanceDot?.y,
      },
      {
        type: "biaxialEffect",
        x: biaxialEffectDot?.x,
        y: biaxialEffectDot?.y,
      },
    ],
    intersection: [
```

```

    {
      type: "burstProduction",
      intersect: burstProductionIntersection?.intersect,
    },
    {
      type: "burstSurfaceAndIntermediate",
      intersect:
burstSurfaceAndIntermediateIntersection?.intersect,
    },
    { type: "collapse", intersect:
collapseIntersection?.intersect },
    { type: "tension", intersect:
tensionIntersection?.intersect },
    { type: "biaxial", intersect:
biaxialIntersection?.intersect },
  ],
  higherThanBounderies: [
    {
      type: "burstProduction",
      higherThanBounderies: burstProductionHigherThanBounderies,
    },
    {
      type: "burstSurfaceAndIntermediate",
      higherThanBounderies:
burstSurfaceAndIntermediateHigherThanBounderies,
    },
    {
      type: "collapse",
      higherThanBounderies: collapseHigherThanBounderies,
    },
    {
      type: "tension",
      higherThanBounderies: tensionHigherThanBounderies,
    },
    {
      type: "biaxial",
      higherThanBounderies: biaxialHigherThanBounderies,
    },
  ],
}
})()

return result
}

```

C.3 Iterasi

```

export const useFindPastBounderies = (
  line1: { x1: number; y1: number; x2: number; y2: number },
  line2: { x1: number; y1: number; x2: number; y2: number }
) => {
  const result = line1.x1 > line2.x1 && line1.x2 > line2.x2

  return result
}

```

C.4 Casing Database

```

export const constant = [
  {
    OD: "4.5",
    NOMINAL_WEIGHT: "9.5",
    GRADE: "J-55",
    COLLAPSE_PRESSURE: "3310",
    IYM_PE: "4380",
    IYM_STC: "4380",
    IYM_LTC: "0",
    IYM_BTC: "0",
    JS_STC: "101",
    JS_LTC: "0",
    JS_BTC: "0",
    BODY_YIELD: "152",
    WALL: "0.205",
    ID: "4.09",
    DRIFT_DIAMETER: "3.965",
    DISPLACEMENT: "0.00342",
    CAPACITY: "0.01625",
  },
  {
    OD: "4.5",
    NOMINAL_WEIGHT: "9.5",
    GRADE: "K-55",
    COLLAPSE_PRESSURE: "3310",
    IYM_PE: "4380",
    IYM_STC: "4380",
    IYM_LTC: "0",
    IYM_BTC: "0",
    JS_STC: "112",
    JS_LTC: "0",
    JS_BTC: "0",
    BODY_YIELD: "152",
    WALL: "0.205",
    ID: "4.09",
    DRIFT_DIAMETER: "3.965",
  }
]

```



```
DISPLACEMENT: "0.00342",
CAPACITY: "0.01625",
},
{
  OD: "4.5",
  NOMINAL_WEIGHT: "9.5",
  GRADE: "LS-65",
  COLLAPSE_PRESSURE: "3600",
  IYM_PE: "5180",
  IYM_STC: "5180",
  IYM_LTC: "0",
  IYM_BTC: "0",
  JS_STC: "135",
  JS_LTC: "0",
  JS_BTC: "0",
  BODY_YIELD: "180",
  WALL: "0.205",
  ID: "4.09",
  DRIFT_DIAMETER: "3.965",
  DISPLACEMENT: "0.00342",
  CAPACITY: "0.01625",
},
// vice versa
```