Summer Fellowship Report

On

Creation of Design report, Debugging, Testing and Descriptive drawing for Osdag Modules

Submitted by

Priti Kumari

Under the guidance of

Prof.Siddhartha Ghosh
Civil Engineering Department
IIT Bombay

Under the Mentorship of

Kumari Anjali Jatav, Yash Lokhande
Project Research Assistant
Osdag,FOSSEE
IIT Bombay

July 3, 2020

Acknowledgment

I would like to thank FOSSEE for providing me a platform to work on something I am very interested in. I am thankful to everyone who thought of having and involved in selection process based on screening tasks. I am grateful to be a part of team which promotes open source software.

I thank all the Osdag members, who are wonderful mentors and great team. I thank Danish Ansari (Assistant Project Manager), Kumari Anjali Jatav (Project Research Assistant), Yash Lokhande (Project Research Assistant), Deepthi Reddy (Project Research Associate), Darshan Viswakarma (Project Research Associate), B Anand Swaroop Goud (Project Research Associate) and Sourabh Das (Project Research Associate), who made us feel welcome and planned all the tasks meticulously during this period.

I am grateful that I got a chance to work under Prof. Sidharth Ghosh, who took time to mentor us and monitored individual contributions as well.

Contents

1 Introduction **3**

1.1 Osdag Internship . . . 3

1.2 What is Osdag? . . . 3

1.3 Who can use ? . . . 3

2 Latex based Design Report **5**

2.1 Beam-Beam and Column-Column Cover Plate Welded Connection 5

2.2 Column-Column End Plate Connection . . . 6

3 Debugging /Testing of Modules **8**

3.1 Beam-Beam and Column-Column Cover Plate Welded Connection 8

3.2 Beam-Beam and Column-Column Cover Plate Bolted Connection 8

4 Descriptive Drawing **10**

4.1 Flange block axial . . . 10

4.2 Web block axial . . . 12

5 Clause References **13**

Appendices **15**

A Design report code of Beam-Beam Cover Plate Welded Connection **16**

B Design report code of Column-Column End Plate Connection **17**

C Excel Sheet for Beam-Beam Cover Plate Welded Connection **19**

D Clause References (Report Function File) **24**

Chapter 1

Introduction

1.1 Osdag Internship

Osdag internship is provided under the FOSSEE project. FOSSEE project promotes the use of FOSS (Free/Libre and Open Source Software) tools to improve quality of education in our country.FOSSEE encourages the use of FOSS tools through various activities to ensure availability of competent free software equivalent to commercial (paid) softwares.
The FOSSEE project is a part of the National Mission on Education through Infrastructure and Communication Technology(ICT), Ministry of Human Resources and Development, Government of India.
Osdag is one such open source software which comes under the FOSSEE project. Osdag internship is provided through FOSSEE project. Any UG/PG/PhD holder can apply for this internship. And the selection will be based on a screening task.

1.2 What is Osdag?

Osdag is Free/Libre and Open Source Software being developed for design of steel structures. Its source code is written in Python, 3D CAD images are developed using PythonOCC. Github is used to ensure smooth workflow between different modules and team members. It is in a path where people from around the world would be able to contribute to its development. FOSSEE's "Share alike" policy would improve the standard of the software when the source code is further modified based on the industrial and educational needs across the country.

1.3 Who can use ?

Osdag is created both for educational purpose and industry professionals. As Osdag is currently funded by MHRD, Osdag team is developing software in

such a way that it can be used by the students during their academics and to give them a better insight look in the subject.
Osdag can be used by anyone starting from novice to professionals. It's simple user interface makes it flexible and attractive than other software. Video tutorials are available to help get started. The video tutorials of Osdag can be accessed here.

Chapter 2

Latex based Design Report

I have created design report for three of the moment connections using LaTex and Pycharm (pylatex package). I have followed Indian Standard codes, various text books, International Standards and INSDAG manuals for creation of these reports. I introduced various functions in report function file which are called to create design report for Beam-Beam Cover Plate Welded Connection, Column-Column Cover Plate Welded Connection, Column-Column End Plate Connection.

2.1 Beam-Beam and Column-Column Cover Plate Welded Connection

Beam-Beam and Column-Column Cover Plate come under the moment connection module. It is used when there is an external moment acting along with axial force and shear force. Some part of design report of Beam-Beam Cover Plate Welded is attached vide Appendix - A

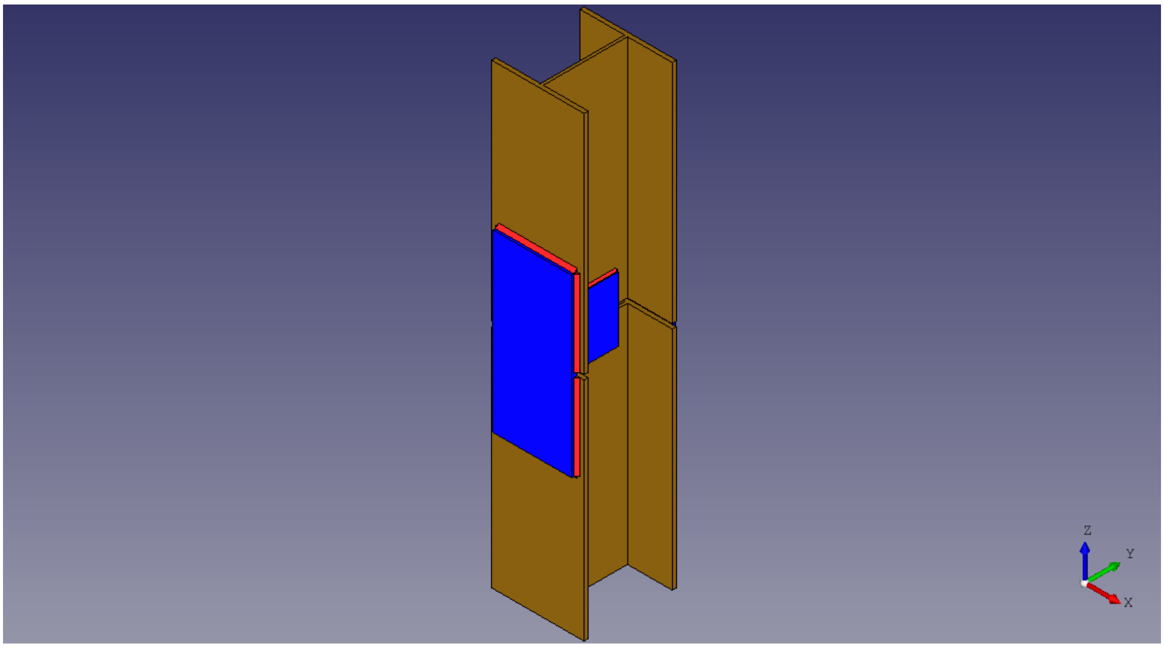

Figure 2.1: 3D drawing output of typical Column-Column Cover Plate Welded connection

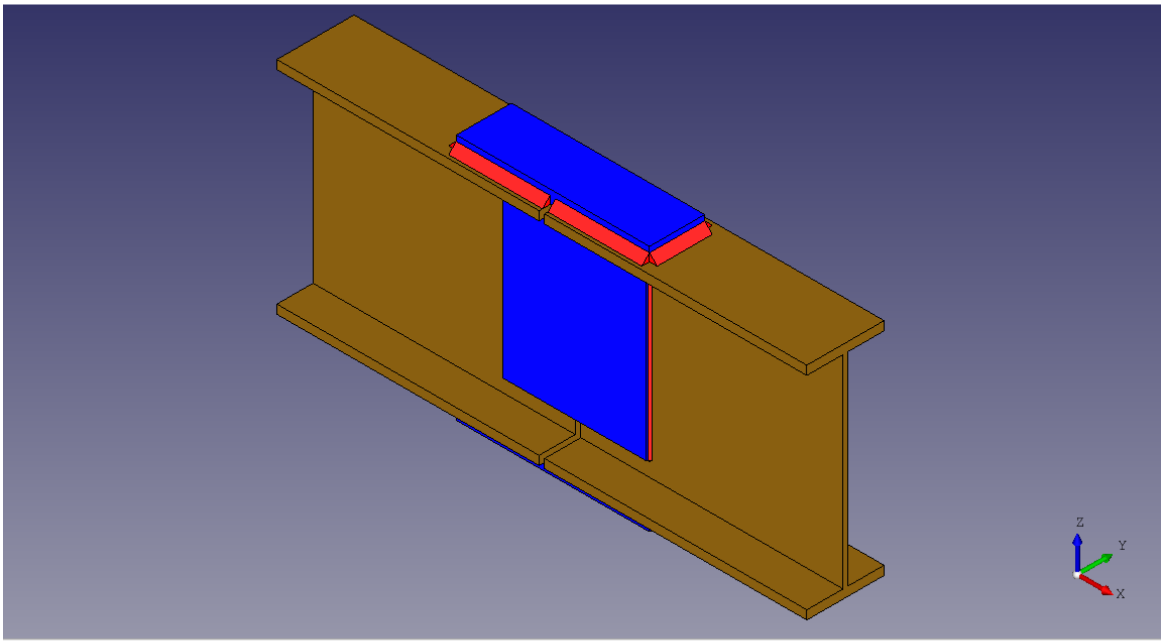

Figure 2.2: 3D drawing output of typical Beam-Beam Cover Plate Welded connection

2.2 Column-Column End Plate Connection

When axial force acting is predominant, then we use Column End Plate Connection which is a sub connection of the Moment Connection Module. Some part of design report of Column-Column End Plate Connection is attached vide Appendix - B

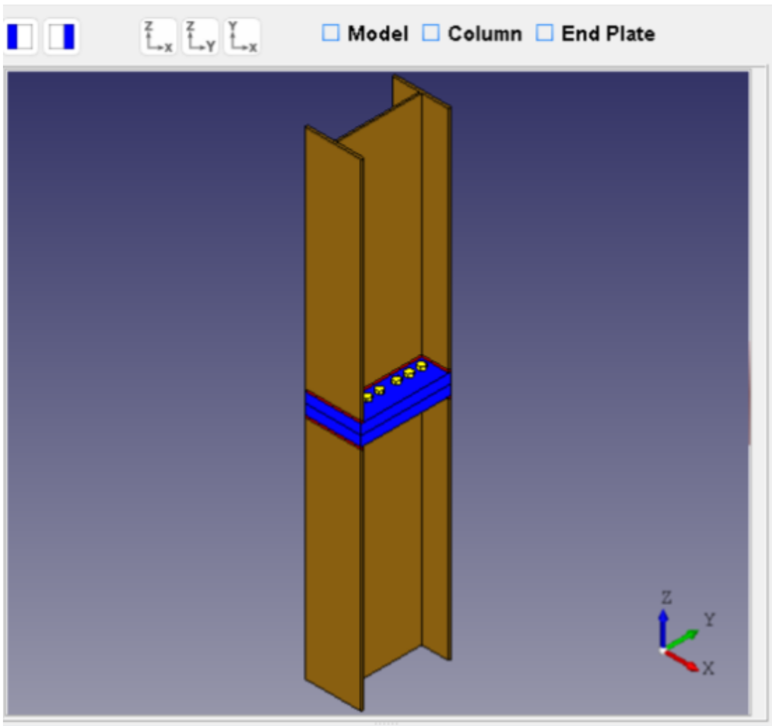

Figure 2.3: 3D drawing output of typical Column-Column End Plate connection

Chapter 3

Debugging /Testing of Modules

I have tested the Osdag modules with various design examples by calculations in MS-Excel for Beam to Beam,Column to Column - Cover Plate Welded Connection. I have checked for crashing report of Beam to Beam, Column to Column - Cover Plate Welded and Bolted Connection both. I have followed the DDCL prepared by Osdag team, various text books, Internationl standards for the calculation.

3.1 Beam-Beam and Column-Column Cover Plate Welded Connection

Beam-Beam and Column-Column Cover Plate come under the moment connection module. It is used when there is an external moment acting along with axial force and shear force. The main reinforcement protruded outward and welded in the beam-beam connection. Connection can transfer a considerable vertical force, horizontal force and moment, when welding and cast concrete are of good quality. To increase the vertical shear distribution, the edges of the beams should be either rough or grooved. The testing done will ensure that the outputs displayed in the output box matches with the outputs from MS Excel. And it will also ensure that the module runs bug free. The Excel sheet of beam-beam welded connection is attached vide Appendix - C

3.2 Beam-Beam and Column-Column Cover Plate Bolted Connection

Beam-Beam and Column-Column Cover Plate come under the moment connection module. It is used when there is an external moment acting along with axial force and shear force. The research and practice using beam-to-beam connections on the precast concrete show more growth. The connection can design according to code for use in areas with high seismic hazards. Installing beam-beam connection that is relatively easier than a beam-column connection, will likely increase its use in the coming years.

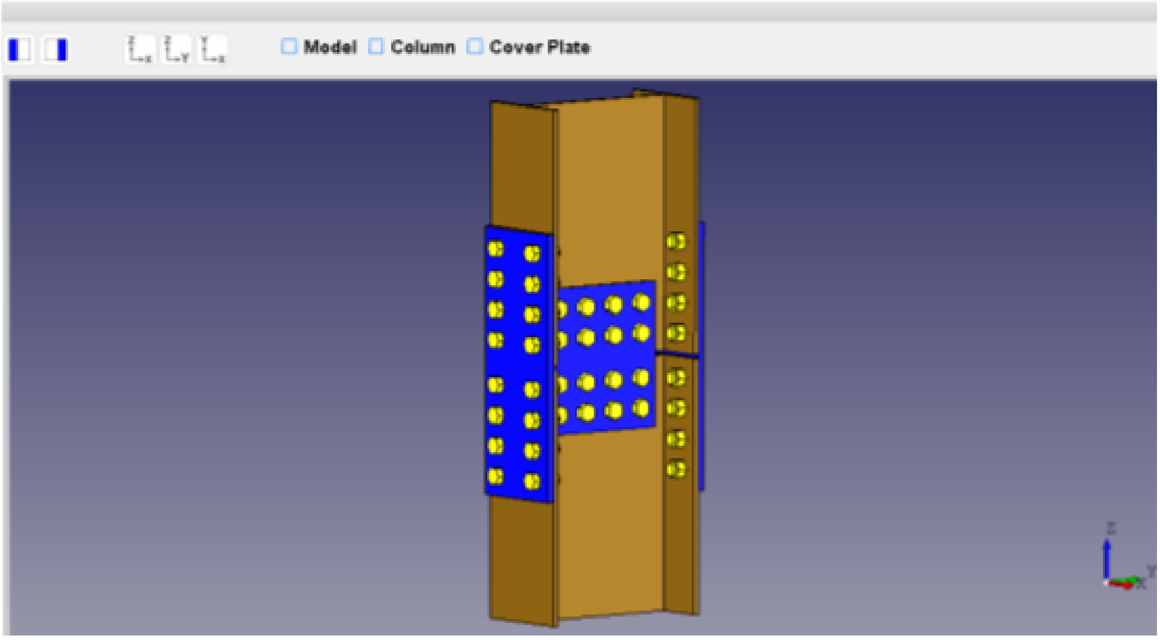

Figure 3.1: 3D drawing output of typical Column-Column Cover Plate Bolted connection

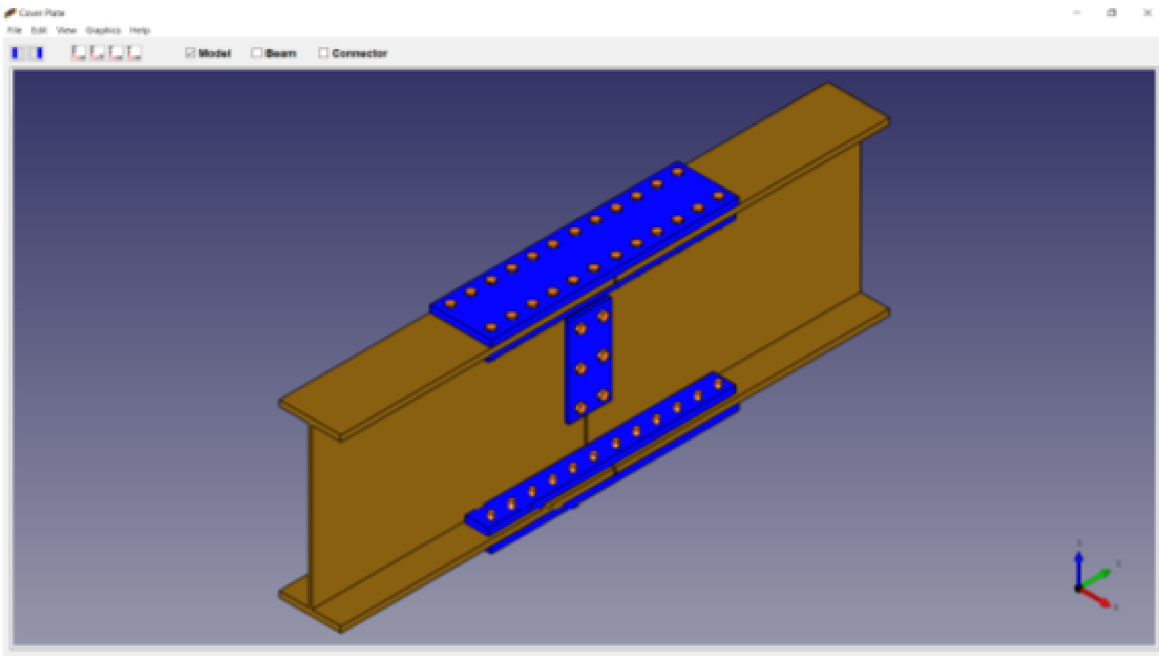

Figure 3.2: 3D drawing output of typical Beam-Beam Cover Plate Bolted connection

Chapter 4

Descriptive Drawing

I have drawn patterns for block shear in flange and web of beam and column both using Inkscape. Inkscape is a cross-platform, professional, free and open-source vector graphics editor. 2D fabrication SVG images, created through Osdag, can be viewed and edited in Inkscape.

4.1 Flange block axial

In the below figure, it shows block shear in flange in axial, block shear in outer flange in axial and block shear in inner flange in axial.

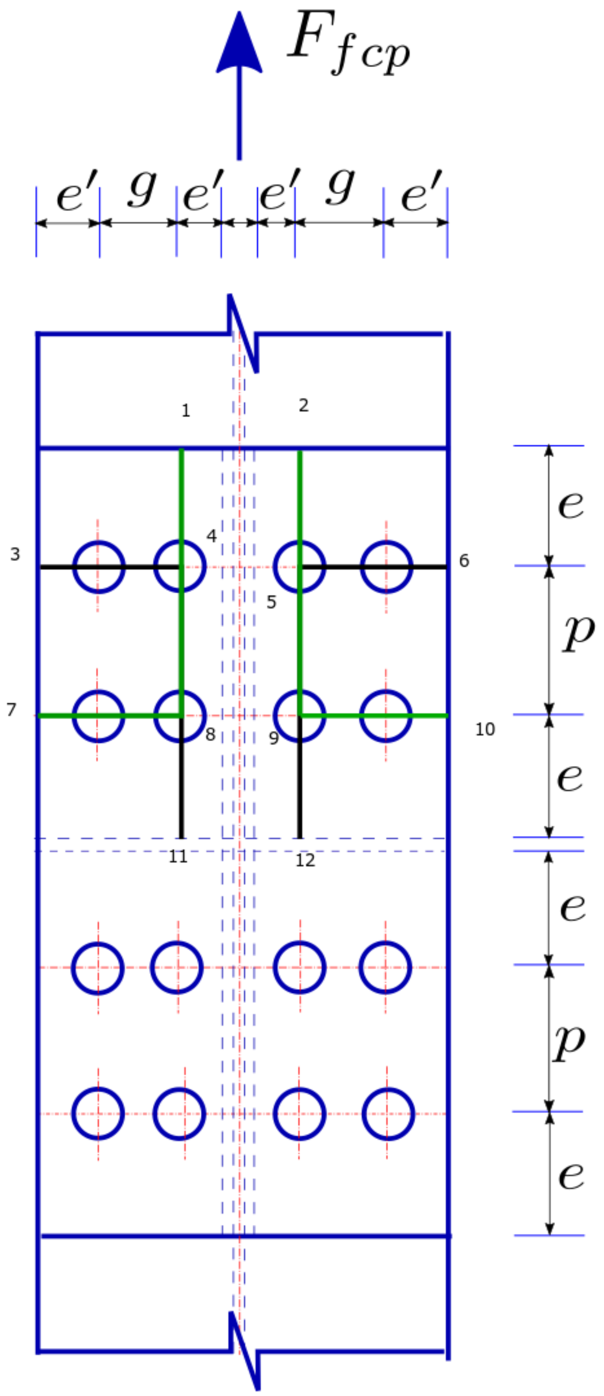

Figure 4.1: 2D drawing of flange block axial (Column)
block shear in flange in axial (3-4-8-11 and 6-5-9-12), block shear in outer flange in axial (1-4-8-7 and 2-5-9-10), block shear in inner flange in axial (1-4-8-7 and 2-5-9-10)

4.2 Web block axial

In the below figure, it shows block shear in web in axial, block shear in web plate in axial and block shear in web plate in shear.

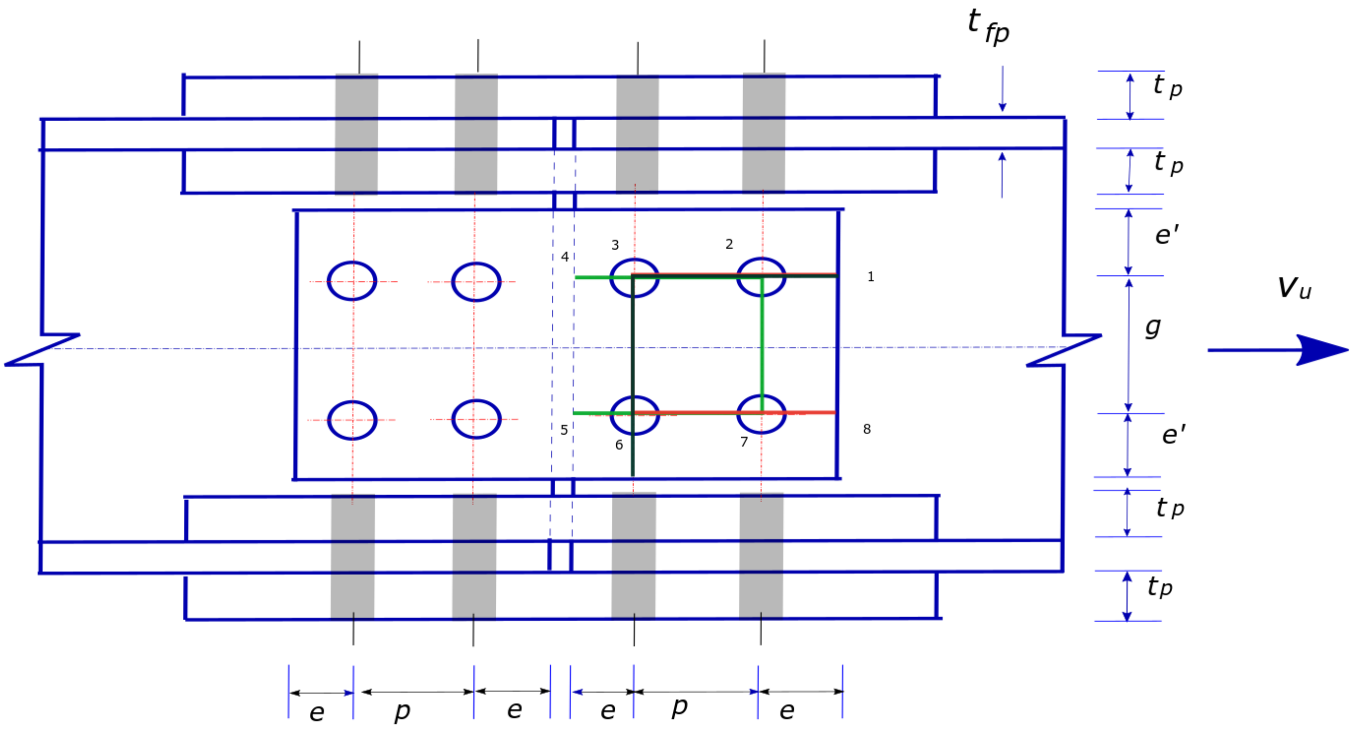

Figure 4.2: 2D drawing of web block axial (Beam)
block shear in web plate in shear (1-2-3-6-9),block shear in web plate in axial (1-2-3-6-7-8),block shear in web in axial (4-3-2-7-6-5)

Chapter 5

Clause References

I put down references for many functions of the Report function file. Anybody can find the references for the functions produced in the 'Report functions.py' file. The clause references of some functions of report function file are attached here Appendix - D

References

IS 800:2007. *Indian Standard Code of Practice for General Construction in Steel*, 2007.

IS 808:1989. *Indian Standard Code of Practice for Dimensions or Hot Rolled Steel Beam, Column, Channel and Angle Sections*, 1989.

SP 6(1):1964. *Indian Standard Code of Practice for Structural Steel Sections*, 1964.

American Institute of Steel Construction. *Steel Construction Manual*, 15 edition, 2017.

American Institute of Steel Construction. *Design guide 4 - Extended End plate moment connections/Seismic and wind applications*, 2016.

American Institute of Steel Construction. *Design guide - 13 - Wide flange column stiffening at moment connections*, 2018a.

American Institute of Steel Construction. *Design guide 16 - Flush and Extended Multiple-Row Moment End-Plate Connections*, 2018b.

Institute of Steel Development and Growth. *Hand Book on Structural Steel Detailing*, 2008.

N. Subramanian. *Design of Steel Structures.* Oxford University Press, 12 edition, 2013.

M.L.Gambhir. *Fundamentals of structural steel design.* McGraw Hill Education, 2013.

Appendices

Appendix A

Design report code of Beam-Beam Cover Plate Welded Connection

```
        if  self.preference == "Outside":
            self.min_height_required = 50
            self.min_length_required = self.flange_plate.height
            t1 = ('SubSection', 'Flange Plate Check',
             ↪  '|p{3.5cm}|p{4.5cm}|p{6cm}|p{1.5cm}|')
            self.report_check.append(t1)
            t1 = (DISP_MIN_FLANGE_PLATE_HEIGHT,self.min_height_required,
             ↪  height_of_flange_cover_plate(B=self.section.flange_width,sp=self.flangespace,b_fp=s
             ↪  self.flange_plate.height, relation="lesser"))
            self.report_check.append(t1)
            t1 =
             ↪  (DISP_MAX_FLANGE_PLATE_HEIGHT,height_of_flange_cover_plate(B=self.section.flange_wi
             ↪  sp=self.flangespace,b_fp=self.flange_plate.height),self.flange_plate.height,get_pas
             ↪  self.flange_plate.height, relation="leq"))
            self.report_check.append(t1)
            t1 = (DISP_MIN_FLANGE_PLATE_LENGTH,
             ↪  self.min_length_required,plate_Length_req(l_w=self.flange_weld.length,
             ↪  t_w=self.flange_weld.size,g=self.flange_plate.gap,
             ↪  l_fp=self.flange_plate.length,conn
             ↪  ="Flange"),get_pass_fail(self.min_length_required,
             ↪  self.flange_plate.length, relation="lesser"))
            self.report_check.append(t1)

            t2 = (DISP_MIN_FLANGE_PLATE_THICK,
             ↪  display_prov(self.section.flange_thickness, "T"),
                  display_prov(self.flange_plate.thickness_provided, "t_{fp}"),
                  get_pass_fail(self.section.flange_thickness,
                   ↪  self.flange_plate.thickness_provided, relation="lesser"))
            self.report_check.append(t2)

```

Appendix B

Design report code of Column-Column End Plate Connection

```
        t1 = ('SubSection', '  End plate Checks', '|p{4cm}|p{6cm}|p{5.5cm}|p{1.5cm}|')
        self.report_check.append(t1)

        if self.connection == "Flush End Plate":

              t1 = (DISP_MIN_PLATE_LENGTH,self.section.depth,
                  self.plate_height,
                  get_pass_fail(self.section.depth, self.plate_height, relation="leq"))
              self.report_check.append(t1)
        else:

              t1 = (DISP_MIN_PLATE_LENGTH, end_plate_ht_req(D=self.section.depth,
              ↪  e=self.end_dist, h_p=self.plate_height),
                  self.plate_height,
                  get_pass_fail(self.plate_height, self.plate_height, relation="leq"))
              self.report_check.append(t1)
        t1 = (DISP_MIN_PLATE_HEIGHT, self.section.flange_width,
                                            self.plate_width,
              get_pass_fail(self.section.flange_width, self.plate_width, relation="leq"))
        self.report_check.append(t1)
        t1 = (DISP_MIN_PLATE_THICK,end_plate_thk_req(M_ep=round(self.m_ep
        ↪  ,2),b_eff=self.b_eff,f_y=self.plate.fy,gamma_m0=gamma_m0,t_p=self.plate_thickness_provided),
               self.plate_thickness_provided,
               get_pass_fail(self.plate.thickness_provided,  self.plate_thickness_provided,
               ↪  relation="leq"))
        self.report_check.append(t1)
        if self.pitch >= 2*self.end_dist:


            ↪  t1=(KEY_OUT_DISP_PLATE_MOM_CAPACITY,moment_acting_on_end_plate(M_ep=round(self.m_ep,
            ↪  2), b_eff=2*self.end_dist, f_y=self.plate.fy, gamma_m0=gamma_m0,
                              t_p=self.plate_thickness_provided),
              design_capacity_of_end_plate(M_dp=round(self.m_dp, 2), b_eff=self.b_eff,
              ↪  f_y=self.plate.fy, gamma_m0=gamma_m0,
                              t_p=self.plate_thickness_provided),
                get_pass_fail(self.m_ep, self.m_dp, relation="leq"))


            self.report_check.append(t1)
        else:
            t1 =
            ↪  (KEY_OUT_DISP_PLATE_MOM_CAPACITY,moment_acting_on_end_plate(M_ep=round(self.m_ep,
            ↪  2), b_eff=self.pitch, f_y=self.plate.fy, gamma_m0=gamma_m0,
```

```
                        t_p=self.plate_thickness_provided),
            design_capacity_of_end_plate(M_dp=round(self.m_dp, 2), b_eff=self.b_eff,
            ↪  f_y=self.plate.fy, gamma_m0=gamma_m0,
                        t_p=self.plate_thickness_provided),
             get_pass_fail(self.m_ep, self.m_dp, relation="leq"))

        self.report_check.append(t1)

    if self.connection == "Extended Both Ways":

            t1 = ('SubSection', '   Stiffener Checks',
            ↪  '|p{4cm}|p{6cm}|p{5.5cm}|p{1.5cm}|')
            self.report_check.append(t1)
            t1 = (KEY_OUT_DISP_STIFFENER_HEIGHT,self.section.depth,
                    self.stiff_ht,
            get_pass_fail(self.section.depth, self.stiff_ht, relation="geq"))
            self.report_check.append(t1)
            t1 = (KEY_OUT_DISP_STIFFENER_WIDTH,end_plate_ht_req(D=self.section.depth,
            ↪  e=self.end_dist, h_p=self.plate_height),

                  self.stiff_wt,
                  get_pass_fail(self.plate_height, self.stiff_wt, relation="geq"))
            self.report_check.append(t1)
            t1 = ( KEY_OUT_DISP_STIFFENER_THICKNESS, '',  self.t_s,'')
            self.report_check.append(t1)
            t1 = (KEY_OUT_DISP_WELD_TYPE, '', self.weld_type, '')
            self.report_check.append(t1)
```

Appendix C

Excel Sheet for Beam-Beam Cover Plate Welded Connection

	A	B	C
1	INPUT PARAMETERS		
2	section	NPB 750*270*197.7	preference
3	fy	250	MPa
4	fu	410	MPa
5	axial load	3	KN
6	shear load	4	KN
7	moment	5	KNm
8	section area	25180	mm^2
9	Gamma Mo	1.1	
10	Depth of section	770	mm
11	flange width	270	mm
12	gamma Mw	1.25	
13	web thk	15.6	mm
14	flange thk	25.4	mm
15	bita_b	1	
16	List thk	[12,14,.....40]	
17	fyw	250	MPa
18	root radius	1.7	
19	Flange weld size	16	mm
20	throat thk(flange)	11.2	mm
21	sp	21	mm
22	yield stress of mat	230	N/mm^2
23	Zp	2017269.696	mm^3
24	Ze	6277640	mm^3
25	Zw	2017269.696	mm^3
26	Z	7212000	mm^3
27	web weld size	8	mm
28	gap	10	mm
29	req eff len	325.00549	

Calculation check for NPB 750*270*197.7 section

D	E	F
MEMBER CAPACITY		
O+I		
Axial capacity	5722.727273	KN
shear capacity	1472.224	KN
plastic moment		
capacity	458.4703855	KNm
moment deformation criteria(kn)		2140.104545
moment capacity	458.4703855	KNm
LOAD CONSIDERATION		
Acmin	1716.818182	KN
Vcmin	883.3344	KN
Mcmin	229.2351927	KNm
Au	1716.818182	KN
Vu	883.3344	KN
Mu	229.2351927	KNm
Aw	764.9672727	KN
Mw	64.11941314	KNm
Af	467.5909091	KN
Mf	165.1157796	KNm
Ff	689.3418889	KN
F.W.strength	2121.016166	N/mm

Member Capacity and Load Consideration

G	H	I	J	K	L
FLANGE WELD DESIGN			WEB WELD DESIGN		
	plate thickness		PLATE thk	8	mm
input thk	18	mm	d	685.8	mm
			ecc	455.4	mm
thk check	TRUE		web pt ht	685.8	mm
Flange area	6858	mm^2	b	669.8	mm
plate ar outside	4104	mm^2	thk check	TRUE	
plate ar outside+inside	7110	mm^2	Web area	11219.52	mm^2
final ar plate	7110	mm^2	Lw_final	685.8	mm
CHECK THE AREA	FAIL		leff _web	2041.4	mm
			Ymax	230.4	mm
if Outside			Xmax	334.9	mm
flange plate ht	228		web pt len	1413.6	mm
flange plate len	530		moment demand	233194949.4	Nmm
Lw_flange	32.50275		web pt area	969446.88	mm^2
l_w_final	228		CHECK THE AREA	PASS	
leff	652				
			Ipw	285549917.7	mm^3
IF I+O			Vex	188.1566515	N/mm
outer pt ht	228		Vey	273.4967994	N/mm
inner pt ht	83.5		Vbh	216.3550505	N/mm
area of flange pt	7110				
otside plate area	4104		Abv	187.3633959	N/mm
area ratio	0.5772151899		web weld stress	613.21	N/mm
weld_eff_length_outer	187.59811		W.W.strength	1060.48	N/mm
l_w flange	-36.200945				
l_w_final	114		COMPARE STRESS N STRENGTH	PASS	
leff	983				Ac

Flange and Web weld design

M	N	O	P	Q	R
MEMBER CHECK		F. PT. CAP. OUT		W.PT. CAP (axial)	
F.T.Y.C(KN)	1558.636364	T.Y.CAP (KN)	858.1090909	T.Y.CAP(KN)	2294.312727
F.T.R.C(KN)	2024.4816	T.R.CAP (KN)	1211.5008	T.R.CAP(KN)	3239.17056
F.tension cap(KN)	1558.636364	F.PT. TEN. CAP(KN)	858.1090909	PT ten cap(KN)	2294.312727
PASS		PASS		PASS	
W.T.Y.C(KN)	2549.890909				
W.T.R.C(KN)	3312.002304	F.PT. CAP(O+I)		W.PT. CAP(shear)	
Avg	21396.96				
Avn	21646.56	T.Y.CAP(KN)	1486.636364	shear Y. cap (KN)	1324.622071
Atg	10698.48	T.R.CAP(KN)	2098.872	shear R. cap (KN)	1870.135995
Atn	10698.48	pt ten cap(KN)	1486.636364	PT shear cap(kn)	1324.622071
Tdb1_web	5965.814163	PASS		PASS	
Tdb2_web	6120.778193				
WEB block shear (kn)	5965.814163				
web tension cap (KN)	2549.890909				
PASS					

Member and Plate Checks

Appendix D

Clause References (Report Function File)

```python

def moment_cap(beta,m_d,f_y,gamma_m0,m_fd,mom_cap):
    """
     Calculate  moment capacity of the column when (class_of_section == 1 or self.class_of_section == 2)

     Args:
            beta: value according to the class of section
            m_d: bending moment acting on the column
            f_y: yield strength of material
            gamma_m0: partial safety factor
            m_fd:factored bending moment acting on the column
            mom_cap: moment capacity of the column
    Returns:
            moment capacity of the column
    """
    #todo reference
    beta= str(beta)
    m_d= str(m_d)
    f_y= str(f_y)
    gamma_m0 = str(gamma_m0)
    m_fd = str(m_fd)
    mom_cap = str(mom_cap)
    moment_cap =Math(inline=True)

    moment_cap.append(NoEscape(r'\begin{aligned} M_{c} &=  m_d - \beta(m_d -m_{fd})  \\'))
    moment_cap.append(NoEscape(r'&= ' + m_d + r'-' + beta + r'('+m_d+r'-'+m_fd+r') \\'))
    moment_cap.append(NoEscape(r'&= ' + mom_cap + r'\end{aligned}'))
    return moment_cap

def moment_CAP( m_d, f_y, gamma_m0, Z_e, mom_cap):
    """
    Calculate  moment capacity of the column
    Args:
            beta: value according to the class of section
            m_d: bending moment acting on the column
            f_y: yield strength of material
            gamma_m0: partial safety factor
            m_fd:factored bending moment acting on the column
            mom_cap: moment capacity of the column
    Returns:
            moment capacity of the column
     Note:
```

```python
            [Ref: cl.8.2.1.2 IS 800:2007]
    """

    m_d = str(m_d)
    f_y = str(f_y)
    gamma_m0 = str(gamma_m0)
    Z_e = str(Z_e)
    mom_cap = str(mom_cap)
    moment_cap = Math(inline=True)

    moment_cap.append(NoEscape(r'\begin{aligned} mom_cap &=  \frac{Z_e*f_y}{\ gamma_m0} \\'))
    moment_cap.append(NoEscape(r'&= \frac{' + Z_e + '*'+f_y+ r'}{' +gamma_m0+r'} \\'))
    moment_cap.append(NoEscape(r'&= ' + mom_cap +r'} \\'))
    moment_cap.append(NoEscape(r'&[Ref: cl.8.2.1.2 IS 800:2007] \end{aligned}'))

    return moment_CAP

def no_of_bolts_along_web(D,T_f,e,p,n_bw):
    """
    Calculate no. of bolts along web

    Args:
        D: section depth in mm  (float)
        T_f:flange thickness in mm  (float)
        e: end distance in mm  (float)
        p:pitch distance in mm  (float)
        n_bw: no. of bolts along web (int)
    Returns:
         no. of bolts along web
    """

    D = str(D)
    e= str(e)
    p = str(p)
    T_f = str(T_f)
    n_bw = str(n_bw)
    no_of_bolts_along_web = Math(inline=True)
    no_of_bolts_along_web.append(NoEscape(r'\begin{aligned} n_{bw} &=  \frac{D -(2*T_f) -(2*e)}{\ p}  + 1 \\'))
    no_of_bolts_along_web.append(NoEscape(r'&= \frac{' + D + ' -(2*'+T_f +')-(2*'+e + r')}{' + p + r'} +1 \\'))
    no_of_bolts_along_web.append(NoEscape(r'&= ' + n_bw + r'\end{aligned}'))
    return no_of_bolts_along_web

def no_of_bolts_along_flange(b,T_w,e,p,n_bf):
    """
    Calculate no of bolts along flange

    Args:
          b:flange width in mm  (float)
          T_w:web thickness in mm  (float)
          e: end distance in mm  (float)
          p: pitch distance in mm  (float)
          n_bf: no. of bolts along flange (int)
    Returns:
          no. of bolts along flange
    """
    b = str(b)
    e= str(e)
    p = str(p)
```

```python
    T_w = str(T_w)
    n_bf = str(n_bf)
    no_of_bolts_along_flange = Math(inline=True)
    no_of_bolts_along_flange.append(NoEscape(r'\begin{aligned} n_{bf} &=  \frac{b/2 -(T_w / 2) -(2*e)}{\ p}  + 1 \\'))
    no_of_bolts_along_flange.append(NoEscape(r'&= \frac{0.5*' + b + ' -(0.5*'+T_w +')-(2*'+e + r')}{' + p + r'} +1 \\'))
    no_of_bolts_along_flange.append(NoEscape(r'&= ' + n_bf + r'\end{aligned}'))
    return no_of_bolts_along_flange


def shear_force_in_bolts_near_web(V,n_wb,V_sb):
    """
    Calculate shear force in each bolts near web

    Args:
           V: factored shear load in KN (float)
           n_wb: no. of bolts in web (int)
           V_sb:shear force in each bolts near web in KN (float)
    Returns:
        shear force in bolts near web
    """
    V = str(V)
    n_wb = str(n_wb)
    V_sb = str(V_sb)
    shear_force_in_bolts_near_web = Math(inline=True)
    shear_force_in_bolts_near_web.append(NoEscape(r'\begin{aligned} V_{sb} &= \frac{V}{\ n_{wb}} \\'))
    shear_force_in_bolts_near_web.append(NoEscape(r'&=\frac{' + V + '}{' + n_wb + r'} \\'))
    shear_force_in_bolts_near_web.append(NoEscape(r'&= ' + V_sb + r'\end{aligned}'))
    return shear_force_in_bolts_near_web

def height_of_flange_cover_plate(B,sp,b_fp): #weld
    """
    Calculate height of falnge cover plate
    Args:
        B:Width of  flange section in mm (float)
        sp: Spacing between flange plate in mm (float)
        b_fp: Height of flange cover plate in mm (float)
    Returns:
          Height of flange cover plate in mm (float)
    """
    B = str(B)
    sp = str(sp)
    b_fp = str (b_fp)
    height_for_flange_cover_plate_eqn =Math(inline=True)

    height_for_flange_cover_plate_eqn.append(NoEscape(r'\begin{aligned} B_{fp} &= {B - 2*sp} \\'))
    height_for_flange_cover_plate_eqn.append(NoEscape(r'&= {' + B + ' - 2 * ' + sp + r'} \\'))

    height_for_flange_cover_plate_eqn.append(NoEscape(r'&=' + b_fp + r'\end{aligned}'))
    return height_for_flange_cover_plate_eqn


def height_of_web_cover_plate(D,sp,b_wp,T,R_1): #weld
    """
    Calculate height of web cover plate
    Args:
        D: Depth of the section in mm (float)
```

```
        sp: Space between web plate in mm (float)
        b_wp: Height of web cover plate in mm (float)
        T: Thickness of flange in mm (float)
        R_1: Root radius in mm (float)
    Returns:
         Height of web cover plate in mm (float)
    """
    D = str(D)
    sp = str(sp)
    b_wp = str (b_wp)
    R_1 = str(R_1)
    T= str(T)
    height_for_web_cover_plate_eqn =Math(inline=True)

    height_for_web_cover_plate_eqn.append(NoEscape(r'\begin{aligned} W_{wp} &= {D-2*T -(2 * R1)- 2*sp} \\'))
    height_for_web_cover_plate_eqn.append(NoEscape(r'&= {' + D + ' - 2 * ' +T+'- (2 *'+ R_1+')- 2 *'+ sp + r'} \\'))

    height_for_web_cover_plate_eqn.append(NoEscape(r'&=' + b_wp + '\end{aligned}'))
    return height_for_web_cover_plate_eqn


def inner_plate_height_weld(B,sp,t,r_1, b_ifp):#weld
    """
    Calculate inner flange plate height for beam welded
    Args:

        B:Width of flange in mm (float)
        sp: Spacing between flange plate in mm (float)
        t: Web thickness in mm (float)
        r_1: Root radius in mm (float)
        b_ifp: Height of inner flange plate in mm (float)
    Returns:
         Height of inner flange plate in mm (float)
    """
    B = str(B)
    sp = str(sp)
    t = str (t)
    r_1 = str(r_1)
    b_ifp = str(b_ifp)
    inner_plate_height_weld_eqn =Math(inline=True)
    inner_plate_height_weld_eqn.append(NoEscape(r'\begin{aligned} B_{ifp} &= \frac{B - 4*sp - t- 2*R1}{2} \\'))
    inner_plate_height_weld_eqn.append(NoEscape(r'&= \frac{'+B +'- 4*'+sp+'-' +t+ '- 2*'+r_1+r'} {2} \\'))
    inner_plate_height_weld_eqn.append(NoEscape(r'&=' + b_ifp + '\end{aligned}'))
    return inner_plate_height_weld_eqn


def plate_Length_req(l_w,t_w,g,l_fp,conn =None): #weld
    """
    Calculate minimum flange plate length
    Args:
       l_w: Weld length of flange in mm (float)
       t_w:Flange weld size in mm (float)
       g: Gap between flange plate in mm (float)
       l_fp: Minimum flange plate length in mm (float)
       conn: Flange or web (str)
    Returns:
          Minimum flange plate length  in mm (float)
```

```
    """
    l_w = str(l_w)
    t_w = str (t_w)
    g = str (g)
    l_fp = str(l_fp)
    min_plate_Length_eqn = Math(inline=True)
    if conn =="Flange":
        min_plate_Length_eqn.append(NoEscape(r'\begin{aligned} L_{fp} & = [2*(l_{w} + 2*t_w) + g]\\'))
        min_plate_Length_eqn.append(NoEscape(r'&= [2*('+ l_w +'+2*'+t_w+') +' + g+ r']\\'))
        min_plate_Length_eqn.append(NoEscape(r'&=' + l_fp + '\end{aligned}'))
    else:
        min_plate_Length_eqn.append(NoEscape(r'\begin{aligned} L_{wp} & = [2*(l_{w} + 2*t_w) + g]\\'))
        min_plate_Length_eqn.append(NoEscape(r'&= [2*(' + l_w + '+2*' + t_w + ') +' + g + r']\\'))
        min_plate_Length_eqn.append(NoEscape(r'&=' + l_fp + '\end{aligned}'))

    return min_plate_Length_eqn


def flange_weld_stress(F_f,l_eff,F_ws):
    """
    Calculate  stress in flange due to welding
    Args:
        F_f: Flange force in KN (float)
        l_eff: Effective weld length of flange in mm (float)
        F_ws: Flange weld stress in KN/mm (float)
    Returns:
         Stress in flange due to welding (float)

      Note:
            Reference:
            IS 800:2007, Cl. 10.5.9
    """
    l_eff = str(l_eff)
    F_ws = str(F_ws)
    F_f =str(F_f)
    flange_weld_stress_eqn = Math(inline=True)
    flange_weld_stress_eqn.append(NoEscape(r'\begin{aligned} Stress &= \frac{F_f*1000}{l_{eff}}}\\'))
    flange_weld_stress_eqn.append(NoEscape(r' &= \frac{' + F_f + '*1000}{' + l_eff + r'}\\'))
    flange_weld_stress_eqn.append(NoEscape(r'&= ' + F_ws + r'\\'))
    flange_weld_stress_eqn.append(NoEscape(r'&[Ref.~IS~800:2007,~Cl.~10.5.9]\end{aligned}'))

    return flange_weld_stress_eqn
```