



MODELAGEM PARAMÉTRICA DOS ESFORÇOS EM VIGAS BIAPOIADAS COMO FERRAMENTA DIDÁTICA NO ENSINO DO COMPORTAMENTO ESTRUTURAL NA ARQUITETURA

DOI: 10.37702/2175-957X.COBENGE.2022.3946

Alexander Lopes de Aquino Brasil - alexander.brasil@ufv.br
Universidade Federal de Viçosa

Resumo: Apresenta-se um algoritmo desenvolvido na plataforma CAD Rhinoceros 3D em conjunto com o editor de programação visual Grasshopper, para análise e visualização dos esforços sobre vigas biapoiadas com carga uniforme distribuída. O método de análise dos esforços implementado no algoritmo se baseia no cálculo das condições de equilíbrio das vigas e construção dos diagramas de esforço cortante e momento fletor. Como resultado, apresenta-se um modelo paramétrico interativo, capaz de produzir cálculos e visualizações instantaneamente conforme a geometria é alterada.

Palavras-chave: modelagem paramétrica; análise de esforços sobre vigas; rhinoceros; grasshopper.



MODELAGEM PARAMÉTRICA DOS ESFORÇOS EM VIGAS BIAPOIADAS COMO FERRAMENTA DIDÁTICA NO ENSINO DO COMPORTAMENTO ESTRUTURAL NA ARQUITETURA

1 INTRODUÇÃO

O processo de projeto arquitetônico envolve, entre muitas demandas, a escolha dos elementos que compõem o sistema estrutural e seus respectivos materiais. Entre as abordagens de ensino das estruturas, pode-se classificar como abordagem qualitativa aquela que utiliza métodos que focam na concepção estrutural a partir do entendimento dos fenômenos físicos ao qual os sistemas estruturais se sujeitam. Por outro lado, classifica-se como abordagem quantitativa o ensino dos cálculos matemáticos das estruturas para análise do comportamento e verificação da segurança (SARAMAGO e LOPES, 2010)

Nesse contexto, Saramago e Lopes (2010) apontam que o ensino de estruturas nos cursos de arquitetura apresenta lacunas. Em seu estudo sobre os currículos de diversas escolas, os autores identificaram não haver a promoção suficiente de integração do ensino de estruturas com as disciplinas de projetos e uma baixa adesão de recursos didáticos que favoreçam o conhecimento qualitativo do comportamento estrutural, priorizando modelos quantitativos e abstratos. Tal abordagem não condiz com as características cognitivas próprias dos estudantes de arquitetura que possuem maior facilidade de compreensão dos conceitos relacionados à concepção estrutural a partir da visualização dos fenômenos físicos que ocorrem nos sistemas estruturais (SARAMAGO e LOPES, 2010).

Atualmente, com a evolução e facilidade de acesso às tecnologias digitais, é importante considerar o seu uso para a demonstração dos conceitos relacionados à concepção de estruturas nos edifícios. Entre as tecnologias computacionais, a modelagem paramétrica se apresenta como um recurso especialmente interessante para o ensino e concepção dos sistemas estruturais durante o processo de projeto. De fato, segundo Silva (2021), uma segunda característica importante deste grupo é a aptidão em lidar com processos lógicos projetuais típicos da modelagem computacional. Estes modelos permitem a visualização dos conceitos estruturais de forma interativa e em tempo real, promovendo assim uma abordagem qualitativa de ensino. Ademais, são baseados em métodos de cálculo que permitem a precisão da análise dos esforços e dimensionamento das estruturas, trazendo precisão ao comportamento estrutural. Por fim, o uso destes modelos auxilia a interpretação dos diagramas geralmente utilizados para visualização do comportamento das estruturas e na compreensão de métodos quantitativos de aprendizagem.

Sendo assim, o presente trabalho apresenta o desenvolvimento de um modelo computacional paramétrico para a análise e visualização dos esforços em um sistema estrutural composto por vigas biapoiadas, submetidas a cargas uniformemente distribuídas, utilizando o programa CAD Rhinoceros 3D em conjunto com o editor de programação visual Grasshopper. O objetivo do trabalho é estabelecer um processo que possa ser aplicado a outros sistemas estruturais vislumbrando o auxílio no aprendizado da concepção estrutural.

2 MODELAGEM PARAMÉTRICA

Segundo Woodburry (2010) a modelagem paramétrica é o desenvolvimento da relação das partes que compõem o projeto de um determinado artefato. Silva (2021) aponta que este processo aplicado ao desenvolvimento de projeto possibilita a concepção dos elementos estruturais paralelamente a forma arquitetônica da edificação em um sistema hierárquico horizontal, em alternativa ao modelo tradicional vertical, onde primeiro se projeta a forma, para depois ser feito o lançamento estrutural e em seguida a escolha dos materiais. Esse modelo facilita a comunicação de uma equipe transdisciplinar de projeto e oferece maior liberdade e autonomia aos arquitetos para exploração de formas geométricas mais complexas ou otimizadas, priorizando a economia de materiais e a sustentabilidade.

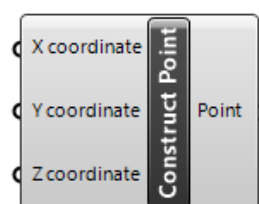
Atualmente, existem uma gama de ferramentas de modelagem paramétrica voltada para arquitetos, engenheiros e outras categorias de projetistas. Dentre estas o programa CAD *Rhinoceros 3D* em conjunto com o editor de programação visual *Grasshopper* se tornou uma das ferramentas mais populares de modelagem paramétrica entre arquitetos e projetistas de outros setores. Testes demonstram que este é um dos ambientes de modelagem paramétrica com bom desempenho de processamento e flexibilidade no desenvolvimento das lógicas projetuais. Além disso, possui versatilidade na conexão com programas de modelagem dos dados da construção (BIM), como *Revit*, *Archicad* e *VisualARQ* (BRITO, SILVA e CHECCUCCI, 2020).

O *Rhinoceros 3D* é um programa CAD desenvolvido pela empresa *Robert McNeel & Associates* que utiliza a tecnologia *NURBS (Non Uniform Rational Basis Splines)* na construção de modelos tridimensionais. Sua tecnologia permite a representação precisa de curvas e superfícies de forma livre, ideais para a modelagem de geometrias complexas.

O *Grasshopper* é um editor de programação visual nativo do *Rhinoceros 3D* que trabalha de forma integrada para a construção de algoritmos que automatizam operações dentro da interface do programa de modelagem 3D. Assim, é possível criar definições precisas para o controle paramétrico sobre modelos gerados em tempo real, explorar processos de projeto generativo e desenvolver lógicas de programação de forma intuitiva baseada na programação visual (AKOS e PARSONS, 2014).

O paradigma da programação visual permite a manipulação da lógica de elementos computacionais graficamente, ao invés de textualmente a partir de blocos de componentes funcionais alimentados por dados de entrada conectados dentro de uma sequência de ações lógicas, e resultando em dados de saída específicos. (AKOS e PARSONS, 2014). conforme representado na Figura 1

Figura 1: Componente funcional do *Grasshopper* para construção de um ponto com base em coordenadas x,y e z.



Fonte: autor.

3 METODOLOGIA

A seguir é apresentada a metodologia utilizada para a construção do algoritmo com base no modelo de análise dos esforços em vigas biapoiadas com carga uniformemente distribuída apresentado por Rebello (2005).

3.1 Cálculo das cargas atuantes

Segundo Rebello (2005), para determinar a carga sobre a laje proveniente do seu peso próprio, determina-se o peso do volume de 1m² de laje:

$$q_{PP\text{laje}} = h * \gamma_{ca} \quad (1)$$

Onde **h** é a altura da laje e γ_{ca} é o peso específico do concreto armado.

O peso do revestimento, para fins de simplificação será considerado 100kgf/m² e o peso proveniente das cargas acidentais é definido conforme tabela fornecida pela NBR 6120. Assim têm que a carga total sobre a laje será:

$$q_{\text{laje}} = q_{PP} + q_{Rv} + q_{Ac} \quad (2)$$

Para o cálculo das cargas provenientes do peso próprio da viga, determina-se o peso do volume de 1m linear de viga:

$$q_{PP\text{viga}} = (b * h) * \gamma_{ca} \quad (3)$$

Onde **h** é a medida da altura da seção transversal da viga, **b** é a medida da base da seção transversal da viga.

No cálculo das cargas proveniente das lajes, quando o lado maior (**L**) for maior que o dobro do lado menor (**I**), considera-se que a laje seja armada em uma direção. Quando **I** for maior ou igual a **L**, então a laje será armada em cruz. Para o caso das lajes armadas em uma só direção, considera-se a carga depositada em 1m linear de viga.

$$q_{Lviga} = q_{\text{laje}} * 2l \quad (4)$$

Para o caso das lajes armadas em cruz, no cálculo da carga incidente sobre as vigas no lado menor, considera-se a carga distribuída sobre a área de influência, neste caso um triângulo:

$$q_{Lviga} = q_{\text{laje}} * \frac{l^2}{4} \quad (5)$$

Já no cálculo da carga incidente sobre as vigas no lado maior, considera-se a carga distribuída sobre a área do trapézio:

$$q_{Lviga} = q_{\text{laje}} * \frac{l^2}{4} * \left(2 - \frac{l}{L}\right) \quad (6)$$

Por fim, considera-se as cargas das alvenarias sobre as lajes. Para isso, calcula-se o peso do volume de uma faixa de alvenaria de 1m de largura ao longo da viga:

$$q_{Aviga} = b * h * \gamma_{ca} \quad (7)$$



Onde **h** é a medida da altura da alvenaria e **b** a medida da espessura da alvenaria. O valor da carga total distribuída sobre a viga será obtido em Kgf/m a partir do somatório de todas as cargas calculadas.

3.2 Cálculo dos esforços atuantes sobre a viga

De acordo com Rebello (2005), na análise dos esforços externos, dada a condição de uma carga uniforme distribuída sobre uma viga biapoiada, considera-se que metade do valor da carga irá ser distribuído em cada apoio (**R1** e **R2**):

$$R1 = R2 = \frac{q * l}{2} \quad (8)$$

Para a obtenção do esforço cortante, considerando o caso específico analisado, a equação para cálculo da força cortante será válida para todo o comprimento da viga. Assim, a análise pode ocorrer em qualquer ponto ao longo do seu comprimento. Tem-se que:

$$Q_x = R - (q_{viga} * x) \quad (9)$$

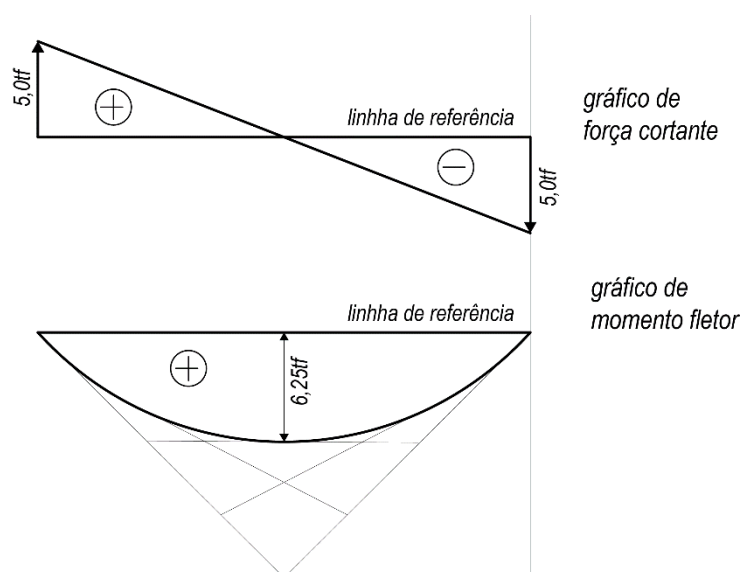
Onde, **R** será a reação de apoio do lado a ser analisado e **x** é a distância do ponto de análise escolhido ao longo do comprimento da viga até **R**. Como **Q_x** depende linearmente da variável **x**, sua representação gráfica será uma reta.

Pela mesma condição anterior da obtenção do esforço interno cortante, a análise do momento fletor pode ser feita em qualquer ponto ao longo do seu comprimento.

$$M_x = (R * x^2) - ((q_{viga} * x) * \frac{x}{2}) \quad (10)$$

Ressalta-se para fins de verificação que o momento fletor máximo ocorrerá onde o esforço cortante passa por zero, neste caso, no centro do comprimento da viga. Como o momento fletor depende do quadrado da variável **x**, sua representação gráfica será uma parábola de segundo grau.

Figura 2: Gráfico de força cortante e momento fletor.



Fonte: (REBELLO, 2005)



3.3 Implementação das cargas atuantes

O objetivo do algoritmo é a análise dos esforços em função da geometria. Logo, os parâmetros de entrada são aqueles que permitem a variação da geometria dos componentes estruturais. Para isso, utiliza-se o componente *Number Slider*, onde valores numéricos dentro de um intervalo são alterados. Valores constantes definidos por tabelas e normas são internalizados em parâmetros primitivos de armazenamento do tipo *Number*. Os parâmetros de entrada do algoritmo podem ser observados na Figura 3.

Figura 3: Parâmetros de entrada para construção da laje e das vigas.

Parâmetros da laje

Dimensões

l1 (m) 7.00

l2 (m) 7.55

h (cm) 17

h do pavimento (m) 4.24

Tipo de uso: Escritório

Peso específico (γca) (Kg/m³)

Peso do revestimento (Kg/m²)

h (m)

h pav (cm)

Carga acidental (Kg/m²)

Parâmetros das vigas

Viga 01

hv1 (cm) 30

bv1 (cm) 20

Há alvenaria sobre a viga? sim

hv1 (m)

bv1 (m)

Viga 02

hv2 (cm) 50

bv2 (cm) 20

Há alvenaria sobre a viga? não

hv2 (m)

bv2 (m)

Viga 03

hv3 (cm) 30

bv3 (cm) 20

Há alvenaria sobre a viga? sim

hv3 (m)

bv3 (m)

Viga 04

hv4 (cm) 50

bv4 (cm) 20

Há alvenaria sobre a viga? não

hv4 (m)

bv4 (m)

l1 (cm)

l2 (cm)

Fonte: autor.

A implementação dos cálculos matemáticos é realizada com o componente *Evaluate* que permite a escrita e avaliação de uma expressão matemática utilizando um número variável de parâmetros de entrada e retornando um parâmetro de saída. De tal forma, ampliando-se o número de variáveis de entrada, implementou-se a Equação (1) para obtenção do peso próprio da laje. O componente pode ser observado na Figura 4. Para aplicar as regras de escolha da armação da laje utiliza-se o componente *Expression* para escrever expressões condicionais do tipo "se, então" (*if, then*), conforme a Figura 5.

sobre a viga, com base no lado correspondente. Em seguida, a carga proveniente do peso próprio da viga e das alvenarias incidentes sobre a viga é calculada implementando-se a Equação (3) e (7), respectivamente.

Para definir se há ou não parede sobre a viga utiliza-se a combinação do componente *Value List* com o uso de uma estrutura condicional utilizando o componente *Expression*. O componente *Value List* permite que o usuário predefina uma lista de valores que podem ser escolhidos de forma interativa, conforme quatro tipos de configurações de escolha. Ainda, o valor escolhido por ser correspondente a outro valor associado. Assim, ao escolher, entre "sim" ou "não" para a pergunta que identifica o componente "Há alvenaria sobre a viga?", o usuário está escolhendo entre 0 ou 1, respectivamente. Estes números são utilizados na estrutura condicional para definir se o componente *Expression* retornará o valor corresponde à carga da alvenaria sobre a viga ou um valor nulo.

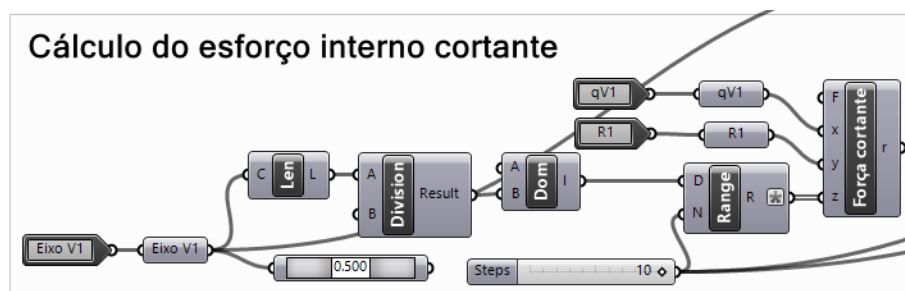
Finalmente, a carga total nas vigas é calculada somando todas as cargas anteriormente encontradas. No entanto, primeiro, é necessário saber se a viga analisada irá utilizar a carga calculada para vigas armadas em uma só direção ou vigas armadas em cruz. Para isso, utiliza-se os valores 0 ou 1 correspondentes ao tipo de armação laje, encontrados a partir do processamento dos componentes descritos na seção 3.1, aplicados em uma estrutura condicional utilizando o componente *Expression*, para escolher corretamente uma entre as duas cargas que são listadas no componente *List Item*. Com a carga proveniente das lajes correta selecionada, utiliza-se o componente *Addition* para somar todas as cargas e retornar o valor total da carga distribuída sobre a viga analisada.

3.4 Implementação da análise dos esforços nas vigas

A análise dos esforços externos ocorre implementando a Equação (8) com o componente *Evaluate*. Obtidos os valores dos esforços externos atuantes sobre a viga, inicia-se a análise dos esforços internos.

Para o cálculo do esforço cortante, primeiro definem-se os pontos ao longo da viga em que será feito a análise dividindo-se o comprimento da viga analisada em duas partes, utilizando o componente *Division*. Esse valor é utilizado em seguida para a construção de um domínio com o componente *Construct Domain*, que será de 0 até a metade do comprimento da viga. Por fim, para definir os pontos, utiliza-se o componente *Range* que gera uma sequência de números com intervalos regulares dentro de um determinado domínio. Utiliza-se o domínio criado com um intervalo de 10 pontos até o centro da viga para definir a localização dos pontos de análise que serão as distâncias utilizadas na análise. Em seguida, utiliza-se o componente *Evaluate* para implementação da Equação (9).

Figura 6: Implementação do algoritmo para cálculo do esforço cortante.



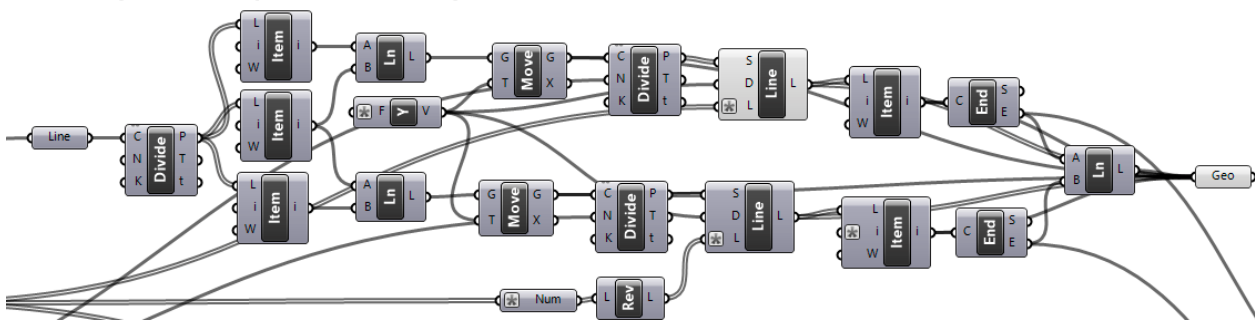
Fonte autor.

Para a construção do diagrama de esforço cortante, é criado um segmento de reta com o comprimento da viga utilizando o componente *Line*. Em seguida, divide-se o segmento ao meio utilizando o componente *Divide Curve*, de onde é possível retornar os pontos da extremidade e o ponto central da reta dividida. Os dois novos segmentos de reta são subdivididos conforme o número definido no componente *Steps*, utilizado para definição das distâncias de análise do esforço cortante ao longo do eixo.

Partindo-se destes pontos criados, utiliza-se o componente *Line SDL* para construção em escala das retas perpendiculares ao eixo da viga, que representam o esforço cortante em seus respectivos pontos. O componente *Line SDL* cria um segmento de reta com base em um ponto inicial, um vetor e um valor de comprimento. Assim, a lista de pontos intervalados ao longo do eixo da viga alimenta a primeira variável. Um vetor pré-definido utilizando o componente *Unit Y* fornece a direção e o sentido de crescimento e o valor dos esforços obtidos define o comprimento de cada linha. Este procedimento é exibido na Figura 7.

Figura 7: Algoritmo para construção do diagrama de esforço cortante.

Construção do diagrama de esforço cortante



Fonte: autor.

Para obter a segunda parcela do gráfico, multiplica-se o vetor por -1 para definir o sentido de crescimento dos segmentos de reta. Finalmente, extrai-se os pontos de extremidades da representação dos esforços localizados nos apoios com a combinação dos componentes *End Points* e *List Item* para construção do eixo diagonal com o componente *Line*.

Na implementação da análise do momento fletor, para obtenção do intervalo de distâncias de análise ao longo do eixo da viga, utiliza-se o mesmo conjunto de componentes da análise do esforço cortante. Em seguida, com o componente *Evaluate* implementa-se a Equação (10). Utilizando o componente *List Item* configurado para retornar sempre o último item da lista com o índice -1, é possível obter o esforço cortante máximo que ocorre no centro da viga. Para construção do diagrama de momento fletor, utiliza-se o mesmo procedimento descrito no diagrama de esforço cortante até o momento de obtenção dos segmentos de retas que representam os esforços utilizando o componente *Line SDL*. Em seguida, são extraídos os pontos de extremidade dos eixos dos esforços. Estes pontos são conectados utilizando o componente *Polyline* para a construção da parábola que configura o diagrama de momento fletor.

3.5 Construção da geometria e posicionamento dos diagramas

Para a representação a geometria da laje e das vigas, utiliza-se o componente *Rectangle*, cujas dimensões x e y são definidas conforme os valores $I1$ e $I2$ respectivamente. O retângulo construído é a base para construção dos eixos de referência da laje e das vigas. Em seguida, utiliza-se o componente *Brep Edges*, que extrai as curvas de uma determinada geometria, para seleção das curvas do retângulo de forma separada, e assim obter os eixos das vigas $V1$, $V2$, $V3$, e $V4$ separadamente com o auxílio do componente *List Item*.

Os eixos são extrudados negativamente em Z com o componente *Extrude* conforme os respectivos valores de altura ($hV1$, $hV2$, $hV3$, $hV4$). Em sequência, utiliza-se o componente *Box Rectangle*, para criar a volumetria da laje, com base no primeiro retângulo, e das vigas com base nos retângulos criados a partir da extrusão. O parâmetro de altura da caixa que representa a laje será a variável de altura da laje (h) e a das caixas que representam as vigas serão as variáveis da base ($bV1$, $bV2$, $bV3$, $bV4$). Neste ponto a geometria que representa as lajes e as vigas está criada. Porém, para uma representação mais didática, representam-se também os pilares que apoiam as vigas e a laje.

Assim, opera-se a translação da geometria das lajes e das vigas na direção Z , em sentido positivo, utilizado o componente *Move* orientado pelo componente *Unit Z* com o valor de altura ($hpav$) definido por um *Number Slider*. Em seguida, são extraídos os vértices de extremidade dos eixos de $V1$ e $V2$ que serão as bases para o posicionamento e construção dos pilares. Novamente, utiliza-se a combinação dos componentes *Rectangle* e *Box Rectangle* para construir a geometria dos pilares.

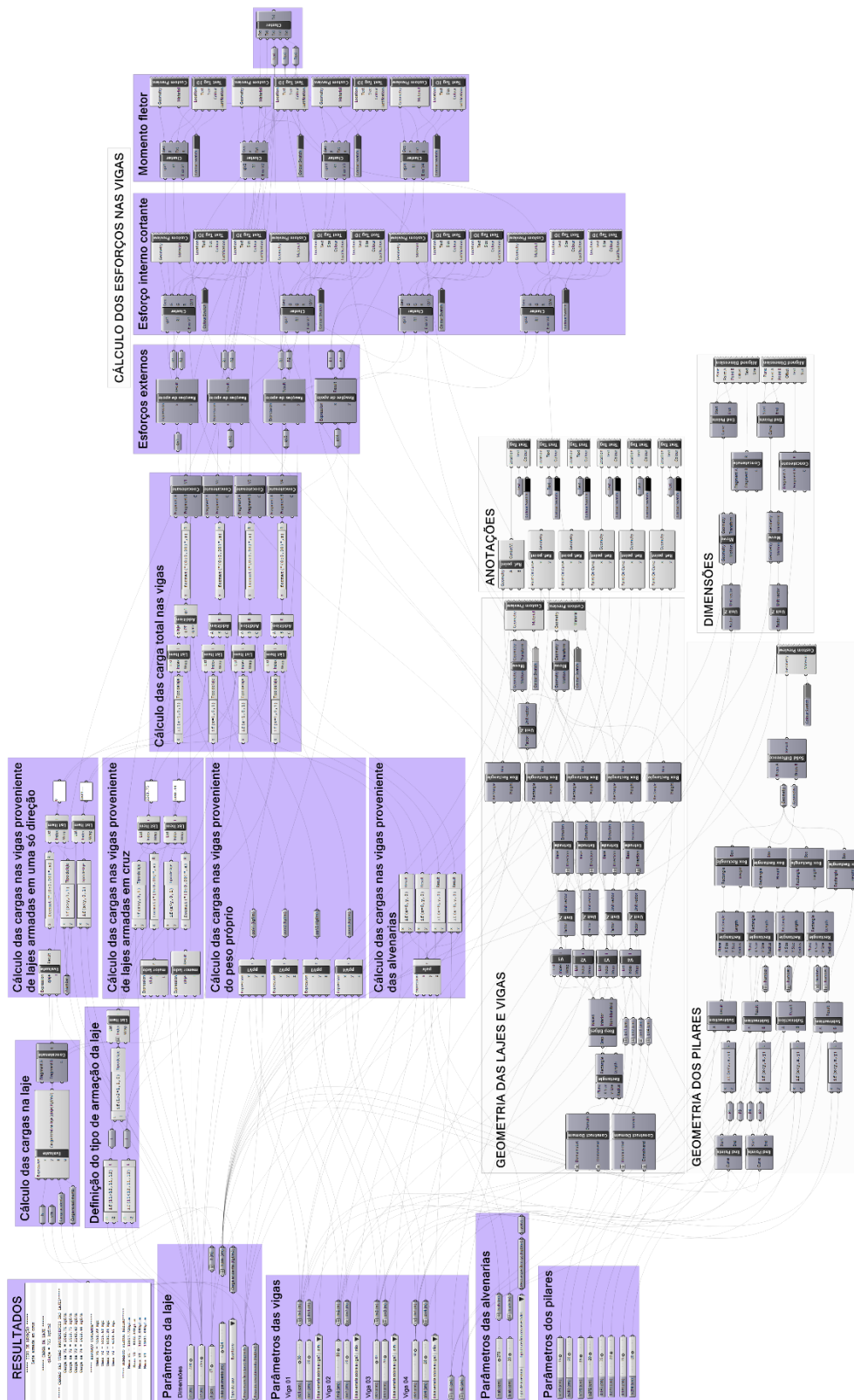
Por fim, os eixos de base das vigas, criados anteriormente são duplicados e posicionados paralelamente à sua referência. Estes novos eixos alimentam o eixo de criação dos diagramas de esforço cortante e momento fletor das vigas. Dessa maneira, cada viga tem seus respectivos diagramas exibidos em tempo real paralelamente ao seu eixo para visualização dos esforços conforme os parâmetros de geometria das lajes, vigas e alvenarias são alterados. Os resultados dos esforços são apresentados juntos ao diagrama e em uma tabela de resultados criada com o componente *Panel*.

Figura 8: Resultados de cálculo operados pelo algoritmo.

RESULTADOS	
Panel	
***** TIPO DE ARMAÇÃO *****	
Laje armada em cruz	
***** CARGA DA LAJE *****	
qlaje = 725 kgf/m ²	
***** CARGAS NAS VIGAS PROVENIENTES DAS LAJES*****	
Carga na V1 = 2043.71 kgf/m	
Carga na V2 = 1518.75 kgf/m	
Carga na V3 = 2143.40 kgf/m	
Carga na V4 = 1618.44 kgf/m	
***** ESFORÇO CORTANTE*****	
Qmax V1 = 7152.99 KgF	
Qmax V2 = 5315.63 KgF	
Qmax V2 = 8091.34 KgF	
Qmax V2 = 6109.61 KgF	
***** MOMENTO FLETOR MÁXIMO*****	
Mmax V1 = 12517.72KgF.m	
Mmax V2 = 9302.34KgF.m	
Mmax V3 = 15272.39KgF.m	
Mmax V4 = 11551.88KgF.m	

Fonte: autor.

Figura 9: Visão geral do script para cálculo das equações e construção dos diagramas em sua versão final.

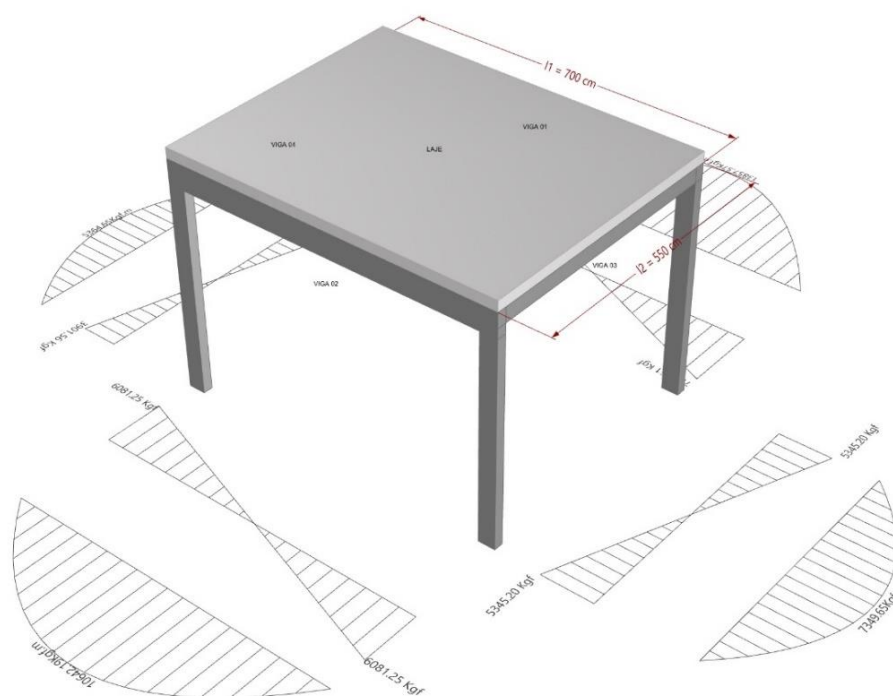


Fonte: autor.

4 RESULTADOS

A partir dos processos descritos foi possível desenvolver um modelo computacional paramétrico capaz calcular as cargas sobre uma viga biapoiada provenientes da laje e das alvenarias, gerar a análise dos esforços externos, esforço interno cortante e do momento fletor, assim como seus respectivos diagramas em aproximadamente 0,1 segundos. O modelo pode ser observado na Figura 10.

Figura 10: Modelo paramétrico desenvolvido.



Fonte: autor.

5 CONCLUSÃO

O presente trabalho utilizou o par de ferramentas *Rhinoceros 3D* e *Grasshopper* para o desenvolvimento de um modelo computacional capaz de auxiliar na visualização e análise automática e em tempo real do comportamento estrutural de um sistema simples de vigas biapoiadas com carga uniformemente distribuída.

É possível concluir que há potencial na ferramenta para o ensino e aprendizado do comportamento das estruturas, uma vez que promove a visualização dos efeitos das cargas sobre a estrutura de forma precisa com base em métodos matemáticos. Ressalva-se que o modelo ainda não foi testado. Destaca-se que código por trás do modelo pode ser reutilizado em outros contextos, ao ser alocado na estrutura de algoritmos mais complexos. Ainda se trata de um primeiro experimento de uma série de modelos a serem desenvolvidos com fins didáticos. Como trabalho futuro pretende-se implementar um novo grupo de funções capazes de dimensionar as peças de forma automatizada com base na análise dos esforços, desenvolver novos modelos de análise e testar o modelo em ambiente prático de ensino e aprendizagem.

6 REFERÊNCIAS

- AKOS, Gil; PARSONS, Ronnie. **FOUNDATIONS: The Grasshopper Primer**. 3^a. ed. Nova York: MODELAB, 2014.
- BRITO, Bruno L. D.; SILVA, Felipe T. D.; CHECCUCCI, Érica D. S. **Estudo Comparativo entre ferramentas de projeto paramétrico**. XVIII Encontro Nacional de Tecnologia do Ambiente Construído ENTAC 2020. Porto Alegre: ANTAC. 2020.
- REBELLO, Yopanan C. P. **Estruturas de Aço, Concreto e Madeira: atendimento da expectativa dimensional**. São Paulo: Ziguarte, 2005.
- SARAMAGO, Rita D. C. P.; LOPES, João M. D. A. Ensino de estruturas nas escolas de arquitetura do Brasil: estrutura curricular e recursos didáticos. **Revista Tecnológica - Edição Especial ENTECA 2009**, Maringá, p. 169-179, Janeiro 2010.
- SILVA, Felipe T. D. Experiências com ferramentas digitais no ensino de estruturas arquitetônicas. **PARC Pesquisa em Arquitetura e Construção**, Campinas, v. 12, n. 00, Agosto 2021. ISSN 1980-6809.
- WOODBURRY, Robert. **Elements of parametric design**. Abingdon: Routledge, 2010.

PARAMETRIC MODELING OF STRESS ANALYSIS IN BI-SUPPORTED BEAMS AS A DIDACTIC TOOL IN THE TEACHING OF STRUCTURAL BEHAVIOR IN ARCHITECTURE

Abstract: This paper presents an algorithm developed with CAD application Rhinoceros 3D and visual programming editor Grasshopper, for stress analysis and visualization of bi-supported beams with a uniform distributed load. The method of analysis implemented in the algorithm is based on the calculation of the equilibrium conditions of the beams and the construction of the diagrams of cutting force and bending moment. As a result, an interactive parametric model is presented, capable of producing calculations and visualizations instantly as the geometry changes.

Keywords: parametric modeling; beams stress analysis; rhinoceros; grasshopper.