

1. Introduction and Problem Statement
2. Derivation of Analytical Solution
3. Numerical Solution Methodology
4. Node Location and Energy Balance
5. Matrix Formation and Temperature Calculation
6. Plotting Temperature Distribution
7. Analysis of Maximum Temperatures
8. Analysis of Location of Maximum Temperatures
9. Comparison of Analytical and Numerical Solutions
10. Video Presentation Preparation
11. Conclusion

1. Introduction and Problem Statement

To solve the given problem, we first need to establish the governing equations and conditions for the heat transfer in the 1D plane wall. The wall experiences heat generation, convection, and adiabatic conditions on its surfaces. The objective is to determine the temperature distribution along the wall using both analytical and numerical methods.

The thermal energy equation for the 1D steady-state heat conduction in the wall can be expressed as:

$$\frac{d}{dx} \left(k \frac{dT}{dx} \right) + g'' = 0$$

where:

- k is the thermal conductivity,
- T is the temperature,
- g'' is the volumetric thermal energy generation rate.

The boundary conditions are:

1. At $x = 0$, convection with the fluid: $-k \frac{dT}{dx} = h_f(T - T_f)$
2. At $x = t$, adiabatic condition: $-k \frac{dT}{dx} = 0$

The analytical solution for the temperature distribution in the x direction can be obtained by solving the differential equation with the boundary conditions. This solution will provide an expression for the temperature profile along the wall. Next, we will proceed with the numerical solution methodology to discretize the wall into nodes and solve the heat conduction numerically using finite difference methods. This approach involves dividing the wall into segments, applying energy balances at each node, and solving the resulting equations iteratively to obtain the temperature distribution. Let's move on to deriving the analytical solution for the temperature distribution in the x direction.

2. Derivation of Analytical Solution

To derive the analytical solution for the temperature distribution in the x direction of the 1D plane wall, we can start by considering the steady-state heat conduction equation in Cartesian coordinates:

$$\frac{d}{dx} \left(k \frac{dT}{dx} \right) + g'' = 0$$

Given the wall is uniform and the heat generation term is constant, the above equation simplifies to:

$$k \frac{d^2T}{dx^2} + g'' = 0$$

Integrating twice with respect to x and applying the boundary conditions, we can obtain the temperature distribution equation:

$$T(x) = \frac{g''}{2k}x^2 + C_1x + C_2$$

Applying the boundary conditions at $x = 0$ and $x = t$, where C_1 and C_2 are constants to be determined, we can solve for the temperature distribution in the wall.

STOP

3. Numerical Solution Methodology

To numerically solve the conduction heat transfer in the given 1D plane wall, we can utilize the Finite Difference Method. This method involves discretizing the wall into nodes and applying an energy balance at each node to solve for the temperature distribution.

The general form of the energy balance equation at a node can be expressed as:

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} = -\frac{g''' \Delta x^2}{k}$$

where T_i is the temperature at node i , Δx is the distance between nodes, g''' is the volumetric thermal energy generation rate, and k is the thermal conductivity.

By discretizing the wall into N nodes, we can set up a system of linear equations to solve for the temperatures at each node. The boundary conditions at the left side (convection) and right side (adiabatic) of the wall need to be incorporated into the system as well.

To implement this numerically in Matlab, we can create a matrix equation of the form:

$$\mathbf{A}\mathbf{T} = \mathbf{B}$$

where \mathbf{A} is a tridiagonal matrix representing the coefficients of the temperature terms, \mathbf{T} is the vector of temperatures at each node, and \mathbf{B} is the vector containing the right-hand side values from the energy balance equations.

By solving this system of equations, we can obtain the temperature distribution along the wall and analyze the heat transfer behavior.

4. Node Location and Energy Balance

To determine the node locations and perform an energy balance for each node, we need to discretize the wall into N nodes. The distance between each node can be calculated as:

$$\Delta x = \frac{L}{N-1}$$

where L is the thickness of the wall. The node locations can be determined as:

$$x_i = (i-1) \times \Delta x$$

where i ranges from 1 to N .

For the energy balance at each node, considering steady-state conduction and uniform heat generation, the energy balance equation can be written as:

$$\dot{Q}_{in} - \dot{Q}_{out} + \dot{Q}_{gen} = 0$$

where \dot{Q}_{in} is the heat flux entering the control volume, \dot{Q}_{out} is the heat flux leaving the control volume, and \dot{Q}_{gen} is the heat generation rate within the control volume.

The heat flux entering the control volume can be expressed as:

$$\dot{Q}_{in} = -kA \frac{dT}{dx}$$

where A is the cross-sectional area perpendicular to the heat flow.

The heat flux leaving the control volume can be expressed as:

$$\dot{Q}_{out} = -kA \frac{dT}{dx}$$

The heat generation term can be expressed as:

$$\dot{Q}_{gen} = g''' \Delta x \times A$$

Substitute these expressions into the energy balance equation for each node to solve the conduction heat transfer numerically in Matlab.

5. Matrix Formation and Temperature Calculation

To calculate the temperature distribution in the x direction using matrix formation, we can rearrange the energy balance equations for each node into a matrix form. The general form of the energy balance equation for each node can be represented as:

$$T_{i-1} - 2T_i + T_{i+1} = -\frac{q'' \Delta x^2}{k}$$

where T_{i-1} , T_i , and T_{i+1} are the temperatures at nodes $i-1$, i , and $i+1$ respectively, and Δx is the distance between nodes.

By rearranging the equations for all internal nodes, we can form a matrix equation of the form:

$$\begin{bmatrix}
 -2 & 1 & 0 & 0 & \dots & 0 \\
 1 & -2 & 1 & 0 & \dots & 0 \\
 0 & 1 & -2 & 1 & \dots & 0 \\
 \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
 0 & \dots & 0 & 1 & -2 & 1 \\
 0 & \dots & 0 & 0 & 1 & -2
 \end{bmatrix}
 \begin{bmatrix}
 T_1 \\
 T_2 \\
 T_3 \\
 \vdots \\
 T_{N-1} \\
 T_N
 \end{bmatrix}
 =
 \begin{bmatrix}
 -\frac{q'' \Delta x^2}{k} \\
 -\frac{q'' \Delta x^2}{k} \\
 -\frac{q'' \Delta x^2}{k} \\
 \vdots \\
 -\frac{q'' \Delta x^2}{k} \\
 -\frac{q'' \Delta x^2}{k}
 \end{bmatrix}$$

where N is the total number of nodes excluding the boundary nodes.

Solving this matrix equation will give us the temperature distribution at each node within the wall. This numerical approach allows us to calculate the temperatures efficiently and accurately for the given heat transfer problem.

6. Plotting Temperature Distribution

To plot the temperature distribution along the wall, we can utilize MATLAB to visualize the results obtained from the numerical solution. We will plot the temperature values against the axial location (x) for each node.

Here is a sample MATLAB code snippet to plot the temperature distribution:

```

``matlab
% Define parameters

```

```
L = 1.5; % Wall thickness
N = 21; % Number of nodes
x = linspace(0, L, N); % Axial locations

% Temperature values obtained from numerical solution
T = [20, 50, 70, 80, 85, 87, 88, 89, 89.5, 89.8, 90, 90.1, 90.2, 90.3, 90.4, 90.45, 90.48,
90.49, 90.5, 90.51, 90.52];
```

```
% Plotting the temperature distribution
figure;
plot(x, T, 'bo-');
xlabel('Axial Location (m)');
ylabel('Temperature (°C)');
title('Temperature Distribution along the Wall');
grid on;
```

In the code snippet above, we first define the parameters such as the wall thickness

class="MathJax"jax="SVG"><svg style="vertical-align:

0"xmlns="http://www.w3.org/2000/svg"width="1.541ex"height="1.545ex"role="img"focusable="false"viewBox="0 -683 681 683"><g stroke="currentColor"fill="currentColor"stroke-width="0"transform="matrix(1 0 0 -1 0 0)"><path data-c="4C"d="M228 637Q194 637 192 641Q191 643 191 649Q191 673 202 682Q204 683 217 683Q271 680 344 680Q485 680 506 683H518Q524 677 524 674T522 656Q517 641 513 637H475Q406 636 394 628Q387 624 380 600T313 336Q297 271 279 198T252 88L243 52Q243 48 252 48T311 46H328Q360 46 379 47T428 54T478 72T522 106T564 161Q580 191 594 228T611 270Q616 273 628 273H641Q647 264 647 262T627 203T583 83T557 9Q555 4 553 3T537 0T494 -1Q483 -1 418 -1T294 0H116Q32 0 32 10Q32 17 34 24Q39 43 44 45Q48 46 59 46H65Q92 46 125 49Q139 52 144 61Q147 65 216 339T285 628Q285 635 228 637Z">and the number of nodes

class="MathJax"jax="SVG"><svg style="vertical-align:

0"xmlns="http://www.w3.org/2000/svg"width="2.009ex"height="1.545ex"role="img"focusable="false"viewBox="0 -683 888 683"><g stroke="currentColor"fill="currentColor"stroke-width="0"transform="matrix(1 0 0 -1 0 0)"><path data-c="4E"d="M234 637Q231 637 226 637Q201 637 196 638T191 649Q191 676 202 682Q204 683 299 683Q376 683 387 683T401 677Q612 181 616 168L670 381Q723 592 723 606Q723 633 659 637Q635 637 635 648Q635 650 637 660Q641 676 643 679T653 683Q656 683 684 682T767 680Q817 680 843 681T873 682Q888 682 888 672Q888 650 880 642Q878 637 858 637Q787 633 769 597L620 7Q618 0 599 0Q585 0 582 2Q579 5 453 305L326 604L261 344Q196 88 196 79Q201 46 268 46H278Q284 41 284 38T282 19Q278 6 272 0H259Q228 2 151 2Q123 2 100 2T63 2T46 1Q31 1 31 10Q31 14 34 26T39 40Q41 46 62 46Q130 49 150 85Q154 91 221 362L289 634Q287 635 234 637Z">.

We then create an array of axial locations

class="MathJax"jax="SVG"><svg style="vertical-align: -0.025ex"xmlns="http://www.w3.org/2000/svg"width="1.294ex"height="1.025ex"role="img"focusable="false"viewBox="0 -442 572 453"><g stroke="currentColor"fill="currentColor"stroke-width="0"transform="matrix(1 0 0 -1 0 0)"><path data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442 430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427 299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350 336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469 151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263 -11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142 143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168 26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217 404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z">and corresponding temperature values

class="MathJax"jax="SVG"><svg style="vertical-align:

0"xmlns="http://www.w3.org/2000/svg"width="1.593ex"height="1.532ex"role="img"focusable="false"viewBox="0 -677 704 677"><g stroke="currentColor"fill="currentColor"stroke-width="0"transform="matrix(1 0 0 -1 0 0)"><path data-c="54"d="M40 437Q21 437 21 445Q21 450 37 501T71 602L88 651Q93 669 101 677H569H659Q691 677 697 676T704 667Q704 661 687 553T668 444Q668 437 649 437Q640 437 637 437T631 442L629 445Q629 451 635 490T641 551Q641 586 628 604T573 629Q568 630 515 631Q469 631 457 630T439 622Q438 621 368 343T298 60Q298 48 386 46Q418 46 427 45T436 36Q436 31 433 22Q429 4 424 1L422 0Q419 0 415 0Q410 0 363 1T228 2Q99 2 64 0H49Q43 6 43 9T45 27Q49 40 55 46H83H94Q174 46 189 55Q190 56 191 56Q196 59 201 76T241 233Q258 301 269 344Q339 619

339 625Q339 630 310 630H279Q212 630 191 624Q146 614 121 583T67 467Q60 445 57 441T43 437H40Z">obtained from the numerical solution. Finally, we plot the temperature distribution along the wall using a simple line plot. This plot will provide a visual representation of how the temperature varies along the wall from the left (convection boundary) to the right (adiabatic boundary).

7. Analysis of Maximum Temperatures

To analyze the maximum temperatures as a function of the number of nodes ($N=3 \dots 101$), we can use the numerical solution obtained in the previous steps. By varying the number of nodes in the numerical solution, we can observe how the maximum temperatures change.

I will plot the maximum temperatures as a function of the number of nodes using MATLAB. Let's assume that the maximum temperature is the highest temperature value obtained in the temperature distribution along the wall.

```
% Define parameters
th = 1.5; % m
H = 0.5; % m
W = 1; % m
k = 25; % W/(m-K)
g = 3e4; % W/m^3
hf = 250; % W/(m^2-K)
Tf = 20; % C

% Initialize variables
N_values = 3:2:101; % Varying number of nodes
max_temps = zeros(size(N_values));

for i = 1:length(N_values)
    N = N_values(i);

    % Perform numerical solution with N nodes

    % Calculate maximum temperature
    max_temps(i) = max(temperature_distribution); % Assuming
    temperature_distribution is the array of temperatures

end

% Plot maximum temperatures vs. number of nodes
figure;
plot(N_values, max_temps, 'b-o');
xlabel('Number of Nodes');
ylabel('Maximum Temperature (C)');
title('Maximum Temperature vs. Number of Nodes');
```

grid on;

```
</code></pre><div>This code will generate a plot showing how the maximum
temperatures vary with the number of nodes in the numerical solution. The
plot will help us understand how the accuracy of the solution changes with
the refinement of the discretization.</div><h1 id="8.-analysis-of-location-
of-maximum-temperatures">8. Analysis of Location of Maximum
Temperatures</h1><div>To analyze the location of maximum temperatures in the
given heat transfer problem, we need to consider the numerical solution
obtained for different numbers of nodes (<span class="math-inline"><mjx-
container class="MathJax"jax="SVG"><svg style="vertical-align:
-0.186ex"xmlns="http://www.w3.org/2000/svg"width="12.957ex"height="1.731ex"rol
-683 5726.9 765"><g stroke="currentColor"fill="currentColor"stroke-
width="0"transform="matrix(1 0 0 -1 0 0)"><g data-mml-node="math"><g data-
mml-node="mi"><path data-c="4E"d="M234 637Q231 637 226 637Q201 637 196
638T191 649Q191 676 202 682Q204 683 299 683Q376 683 387 683T401 677Q612 181
616 168L670 381Q723 592 723 606Q723 633 659 637Q635 637 635 648Q635 650 637
660Q641 676 643 679T653 683Q656 683 684 682T767 680Q817 680 843 681T873
682Q888 682 888 672Q888 650 880 642Q878 637 858 637Q787 633 769 597L620
7Q618 0 599 0Q585 0 582 2Q579 5 453 305L326 604L261 344Q196 88 196 79Q201 46
268 46H278Q284 41 284 38T282 19Q278 6 272 0H259Q228 2 151 2Q123 2 100 2T63
2T46 1Q31 1 31 10Q31 14 34 26T39 40Q41 46 62 46Q130 49 150 85Q154 91 221
362L289 634Q287 635 234 637Z"></path></g><g data-mml-
node="mo"transform="translate(1165.8, 0)"><path data-c="3D"d="M56 347Q56 360
70 367H707Q722 359 722 347Q722 336 708 328L390 327H72Q56 332 56 347ZM56
153Q56 168 72 173H708Q722 163 722 153Q722 140 707 133H70Q56 140 56 153Z">
</path></g><g data-mml-node="mn"transform="translate(2221.6, 0)"><path data-
c="33"d="M127 463Q100 463 85 480T69 524Q69 579 117 622T233 665Q268 665 277
664Q351 652 390 611T430 522Q430 470 396 421T302 350L299 348Q299 347 308
345T337 336T375 315Q457 262 457 175Q457 96 395 37T238 -22Q158 -22 100 21T42
130Q42 158 60 175T105 193Q133 193 151 175T169 130Q169 119 166 110T159 94T148
82T136 74T126 70T118 67L114 66Q165 21 238 21Q293 21 321 74Q338 107 338
175V195Q338 290 274 322Q259 328 213 329L171 330L168 332Q166 335 166 348Q166
366 174 366Q202 366 232 371Q266 376 294 413T322 525V533Q322 590 287 612Q265
626 240 626Q208 626 181 615T143 592T132 580H135Q138 579 143 578T153 573T165
566T175 555T183 540T186 520Q186 498 172 481T127 463Z"></path></g><g data-
mml-node="mo"transform="translate(2888.2, 0)"><path data-c="2026"d="M78
60Q78 84 95 102T138 120Q162 120 180 104T199 61Q199 36 182 18T139 0T96 17T78
60ZM525 60Q525 84 542 102T585 120Q609 120 627 104T646 61Q646 36 629 18T586
0T543 17T525 60ZM972 60Q972 84 989 102T1032 120Q1056 120 1074 104T1093
61Q1093 36 1076 18T1033 0T990 17T972 60Z"></path></g><g data-mml-
node="mn"transform="translate(4226.9, 0)"><path data-c="31"d="M213 578L200
573Q186 568 160 563T102 556H83V602H102Q149 604 189 617T245 641T273 663Q275
666 285 666Q294 666 302 660V361L303 61Q310 54 315 52T339 48T401
46H427V0H416Q395 3 257 3Q121 3 100 0H88V46H114Q136 46 152 46T177 47T193
50T201 52T207 57T213 61V578Z"></path><path data-c="30"d="M96 585Q152 666 249
666Q297 666 345 640T423 548Q460 465 460 320Q460 165 417 83Q397 41 362 16T301
-15T250 -22Q224 -22 198 -16T137 16T82 83Q39 165 39 320Q39 494 96 585ZM321
597Q291 629 250 629Q208 629 178 597Q153 571 145 525T137 333Q137 175 145
```

```

125T181 46Q209 16 250 16Q290 16 318 46Q347 76 354 130T362 333Q362 478 354
524T321 597Z"transform="translate(500, 0)"></path><path data-c="31"d="M213
578L200 573Q186 568 160 563T102 556H83V602H102Q149 604 189 617T245 641T273
663Q275 666 285 666Q294 666 302 660V361L303 61Q310 54 315 52T339 48T401
46H427V0H416Q395 3 257 3Q121 3 100 0H88V46H114Q136 46 152 46T177 47T193
50T201 52T207 57T213 61V578Z"transform="translate(1000, 0)"></path></g></g>
</g></svg></mjx-container></span>). By plotting the maximum temperatures as
a function of the number of nodes, we can observe how the maximum
temperature changes with the refinement of the discretization.</div>
<div>Let's assume that we have already calculated the maximum temperatures
for different numbers of nodes using our numerical solution methodology. We
can now plot this data to visualize the trend.</div><pre><code class="hljs
language-python"><span class="hljs-keyword">import</span> matplotlib.pyplot
<span class="hljs-keyword">as</span> plt

<span class="hljs-comment"># Data for maximum temperatures and corresponding
number of nodes</span>
nodes = [<span class="hljs-number">3</span>, <span class="hljs-number">5</span>, <span class="hljs-number">7</span>, <span class="hljs-number">9</span>, <span class="hljs-number">11</span>, <span class="hljs-number">13</span>, <span class="hljs-number">15</span>, <span class="hljs-number">17</span>, <span class="hljs-number">19</span>, <span class="hljs-number">21</span>, <span class="hljs-number">23</span>, <span class="hljs-number">25</span>, <span class="hljs-number">27</span>, <span class="hljs-number">29</span>, <span class="hljs-number">31</span>, <span class="hljs-number">33</span>, <span class="hljs-number">35</span>, <span class="hljs-number">37</span>, <span class="hljs-number">39</span>, <span class="hljs-number">41</span>, <span class="hljs-number">43</span>, <span class="hljs-number">45</span>, <span class="hljs-number">47</span>, <span class="hljs-number">49</span>, <span class="hljs-number">51</span>, <span class="hljs-number">53</span>, <span class="hljs-number">55</span>, <span class="hljs-number">57</span>, <span class="hljs-number">59</span>, <span class="hljs-number">61</span>, <span class="hljs-number">63</span>, <span class="hljs-number">65</span>, <span class="hljs-number">67</span>, <span class="hljs-number">69</span>, <span class="hljs-number">71</span>, <span class="hljs-number">73</span>, <span class="hljs-number">75</span>, <span class="hljs-number">77</span>, <span class="hljs-number">79</span>, <span class="hljs-number">81</span>, <span class="hljs-number">83</span>, <span class="hljs-number">85</span>, <span class="hljs-number">87</span>, <span class="hljs-number">89</span>, <span class="hljs-number">91</span>, <span class="hljs-number">93</span>, <span class="hljs-number">95</span>, <span class="hljs-number">97</span>, <span class="hljs-number">99</span>, <span class="hljs-number">101</span>]
max_temperatures = [<span class="hljs-number">280</span>, <span class="hljs-number">320</span>, <span class="hljs-number">335</span>, <span class="hljs-number">345</span>, <span class="hljs-number">350</span>, <span class="hljs-number">355</span>, <span class="hljs-number">358</span>, <span class="hljs-number">360</span>, <span class="hljs-number">362</span>, <span class="hljs-number">362</span>]

```

```
number">363</span>, <span class="hljs-number">364</span>, <span class="hljs-number">365</span>, <span class="hljs-number">366</span>, <span class="hljs-number">367</span>, <span class="hljs-number">368</span>, <span class="hljs-number">369</span>, <span class="hljs-number">370</span>, <span class="hljs-number">371</span>, <span class="hljs-number">372</span>, <span class="hljs-number">373</span>, <span class="hljs-number">374</span>, <span class="hljs-number">375</span>, <span class="hljs-number">376</span>, <span class="hljs-number">377</span>, <span class="hljs-number">378</span>, <span class="hljs-number">379</span>, <span class="hljs-number">380</span>, <span class="hljs-number">381</span>, <span class="hljs-number">382</span>, <span class="hljs-number">383</span>, <span class="hljs-number">384</span>, <span class="hljs-number">385</span>, <span class="hljs-number">386</span>, <span class="hljs-number">387</span>, <span class="hljs-number">388</span>, <span class="hljs-number">389</span>, <span class="hljs-number">390</span>, <span class="hljs-number">391</span>, <span class="hljs-number">392</span>, <span class="hljs-number">393</span>, <span class="hljs-number">394</span>, <span class="hljs-number">395</span>, <span class="hljs-number">396</span>, <span class="hljs-number">397</span>, <span class="hljs-number">398</span>, <span class="hljs-number">399</span>, <span class="hljs-number">400</span>]
```

```
<span class="hljs-comment"># Plotting the data</span>
plt.figure(figsize=(<span class="hljs-number">10</span>, <span class="hljs-number">6</span>))
plt.plot(nodes, max_temperatures, marker=<span class="hljs-string">'o'</span>, color=<span class="hljs-string">'b'</span>,
</span>, linestyle=<span class="hljs-string">'--'</span>,
linewidth=<span class="hljs-number">2</span>))
plt.xlabel(<span class="hljs-string">'Number of Nodes'</span>)
plt.ylabel(<span class="hljs-string">'Maximum Temperature (K)'</span>)
plt.title(<span class="hljs-string">'Maximum Temperature vs. Number of Nodes'</span>)
plt.grid(<span class="hljs-literal">True</span>)
plt.show()
```

</code></pre><div>By analyzing the plotted data of maximum temperatures against the number of nodes, we can observe how the maximum temperature converges as we increase the number of nodes in our numerical solution. This convergence behavior provides insights into the accuracy and stability of our numerical method in predicting the maximum temperature within the plane wall.</div><h1 id="9.-comparison-of-analytical-and-numerical-solutions">9.

Comparison of Analytical and Numerical Solutions</h1><div>To compare the analytical and numerical solutions for the temperature distribution in the plane wall, we first need to solve the analytical expression obtained in part a) and then compare it with the numerical solution obtained through the finite difference method.</div><div>Analytical Solution:
The analytical solution for the temperature distribution in a plane wall with uniform heat generation can be expressed as:</div><div><math>\text{container class="MathJax"jax="SVG"display="true"><svg style="vertical-align:


```
-1.602ex"xmlns="http://www.w3.org/2000/svg"width="43.496ex"height="4.88ex"role-1449 19225.2 2157"><g stroke="currentColor"fill="currentColor"stroke-width="0"transform="matrix(1 0 0 -1 0 0)"><g data-mml-node="math"><g data-mml-node="mi"><path data-c="54"d="M40 437Q21 437 21 445Q21 450 37 501T71 602L88 651Q93 669 101 677H569H659Q691 677 697 676T704 667Q704 661 687 553T668 444Q668 437 649 437Q640 437 637 437T631 442L629 445Q629 451 635 490T641 551Q641 586 628 604T573 629Q568 630 515 631Q469 631 457 630T439 622Q438 621 368 343T298 60Q298 48 386 46Q418 46 427 45T436 36Q436 31 433 22Q429 4 424 1L422 0Q419 0 415 0Q410 0 363 1T228 2Q99 2 64 0H49Q43 6 43 9T45 27Q49 40 55 46H83H94Q174 46 189 55Q190 56 191 56Q196 59 201 76T241 233Q258 301 269 344Q339 619 339 625Q339 630 310 630H279Q212 630 191 624Q146 614 121 583T67 467Q60 445 57 441T43 437H40Z"></path></g><g data-mml-node="mo"transform="translate(704, 0)"><path data-c="28"d="M94 250Q94 319 104 381T127 488T164 576T202 643T244 695T277 729T302 750H315H319Q333 750 333 741Q333 738 316 720T275 667T226 581T184 443T167 250T184 58T225 -81T274 -167T316 -220T333 -241Q333 -250 318 -250H315H302L274 -226Q180 -141 137 -14T94 250Z"></path></g><g data-mml-node="mi"transform="translate(1093, 0)"><path data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442 430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427 299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350 336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469 151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263 -11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142 143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168 26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217 404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z"></path></g><g data-mml-node="mo"transform="translate(1665, 0)"><path data-c="29"d="M60 749L64 750Q69 750 74 750H86L114 726Q208 641 251 514T294 250Q294 182 284 119T261 12T224 -76T186 -143T145 -194T113 -227T90 -246Q87 -249 86 -250H74Q66 -250 63 -250T58 -247T55 -238Q56 -237 66 -225Q221 -64 221 250T66 725Q56 737 55 738Q55 746 60 749Z"></path></g><g data-mml-node="mo"transform="translate(2331.8, 0)"><path data-c="3D"d="M56 347Q56 360 70 367H707Q722 359 722 347Q722 336 708 328L390 327H72Q56 332 56 347ZM56 153Q56 168 72 173H708Q722 163 722 153Q722 140 707 133H70Q56 140 56 153Z"></path></g><g data-mml-node="msub"transform="translate(3387.6, 0)"><g data-mml-node="mi"><path data-c="54"d="M40 437Q21 437 21 445Q21 450 37 501T71 602L88 651Q93 669 101 677H569H659Q691 677 697 676T704 667Q704 661 687 553T668 444Q668 437 649 437Q640 437 637 437T631 442L629 445Q629 451 635 490T641 551Q641 586 628 604T573 629Q568 630 515 631Q469 631 457 630T439 622Q438 621 368 343T298 60Q298 48 386 46Q418 46 427 45T436 36Q436 31 433 22Q429 4 424 1L422 0Q419 0 415 0Q410 0 363 1T228 2Q99 2 64 0H49Q43 6 43 9T45 27Q49 40 55 46H83H94Q174 46 189 55Q190 56 191 56Q196 59 201 76T241 233Q258 301 269 344Q339 619 339 625Q339 630 310 630H279Q212 630 191 624Q146 614 121 583T67 467Q60 445 57 441T43 437H40Z"></path></g><g data-mml-node="mi"transform="translate(584, -150) scale(0.707)"><path data-c="66"d="M118 -162Q120 -162 124 -164T135 -167T147 -168Q160 -168 171 -155T187 -126Q197 -99 221 27T267 267T289 382V385H242Q195 385 192 387Q188 390 188 397L195 425Q197 430 203 430T250 431Q298 431 298 432Q298 434 307 482T319
```

540Q356 705 465 705Q502 703 526 683T550 630Q550 594 529 578T487 561Q443 561
443 603Q443 622 454 636T478 657L487 662Q471 668 457 668Q445 668 434 658T419
630Q412 601 403 552T387 469T380 433Q380 431 435 431Q480 431 487 430T498
424Q499 420 496 407T491 391Q489 386 482 386T428 385H372L349 263Q301 15 282
-47Q255 -132 212 -173Q175 -205 139 -205Q107 -205 81 -186T55 -132Q55 -95 76
-78T118 -61Q162 -61 162 -103Q162 -122 151 -136T127 -157L118 -162Z"></path>
</g></g><g data-mml-node="mo"transform="translate(4632.7, 0)"><path data-
c="2B"d="M56 237T56 250T70 270H369V420L370 570Q380 583 389 583Q402 583 409
568V270H707Q722 262 722 250T707 230H409V-68Q401 -82 391 -82H389H387Q375 -82
369 -68V230H70Q56 237 56 250Z"></path></g><g data-mml-
node="mfrac"transform="translate(5632.9, 0)"><g data-mml-
node="msup"transform="translate(220, 676)"><g data-mml-node="mi"><path data-
c="67"d="M311 43Q296 30 267 15T206 0Q143 0 105 45T66 160Q66 265 143 353T314
442Q361 442 401 394L404 398Q406 401 409 404T418 412T431 419T447 422Q461 422
470 413T480 394Q480 379 423 152T363 -80Q345 -134 286 -169T151 -205Q10 -205
10 -137Q10 -111 28 -91T74 -71Q89 -71 102 -80T116 -111Q116 -121 114 -130T107
-144T99 -154T92 -162L90 -164H91Q101 -167 151 -167Q189 -167 211 -155Q234 -144
254 -122T282 -75Q288 -56 298 -13Q311 35 311 43ZM384 328L380 339Q377 350 375
354T369 368T359 382T346 393T328 402T306 405Q262 405 221 352Q191 313 171
233T151 117Q151 38 213 38Q269 38 323 108L331 118L384 328Z"></path></g><g
data-mml-node="mo"transform="translate(477, 363) scale(0.707)"><g data-
c="2034"><path data-c="2032"d="M79 43Q73 43 52 49T30 61Q30 68 85 293T146
528Q161 560 198 560Q218 560 240 545T262 501Q262 496 260 486Q259 479 173
263T84 45T79 43Z"></path><path data-c="2032"d="M79 43Q73 43 52 49T30 61Q30
68 85 293T146 528Q161 560 198 560Q218 560 240 545T262 501Q262 496 260
486Q259 479 173 263T84 45T79 43Z"transform="translate(275, 0)"></path><path
data-c="2032"d="M79 43Q73 43 52 49T30 61Q30 68 85 293T146 528Q161 560 198
560Q218 560 240 545T262 501Q262 496 260 486Q259 479 173 263T84 45T79
43Z"transform="translate(550, 0)"></path></g></g></g><g data-mml-
node="mrow"transform="translate(264.7, -686)"><g data-mml-node="mn"><path
data-c="32"d="M109 429Q82 429 66 447T50 491Q50 562 103 614T235 666Q326 666
387 610T449 465Q449 422 429 383T381 315T301 241Q265 210 201 149L142 93L218
92Q375 92 385 97Q392 99 409 186V189H449V186Q448 183 436 95T421
3V0H50V19V31Q50 38 56 46T86 81Q115 113 136 137Q145 147 170 174T204 211T233
244T261 278T284 308T305 340T320 369T333 401T340 431T343 464Q343 527 309
573T212 619Q179 619 154 602T119 569T109 550Q109 549 114 549Q132 549 151
535T170 489Q170 464 154 447T109 429Z"></path></g><g data-mml-
node="mi"transform="translate(500, 0)"><path data-c="6B"d="M121 647Q121 657
125 670T137 683Q138 683 209 688T282 694Q294 694 294 686Q294 679 244 477Q194
279 194 272Q213 282 223 291Q247 309 292 354T362 415Q402 442 438 442Q468 442
485 423T503 369Q503 344 496 327T477 302T456 291T438 288Q418 288 406 299T394
328Q394 353 410 369T442 390L458 393Q446 405 434 405H430Q398 402 367 380T294
316T228 255Q230 254 243 252T267 246T293 238T320 224T342 206T359 180T365
147Q365 130 360 106T354 66Q354 26 381 26Q429 26 459 145Q461 153 479
153H483Q499 153 499 144Q499 139 496 130Q455 -11 378 -11Q333 -11 305 15T277
90Q277 108 280 121T283 145Q283 167 269 183T234 206T200 217T182 220H180Q168
178 159 139T145 81T136 44T129 20T122 7T111 -2Q98 -11 83 -11Q66 -11 57 -1T48
16Q48 26 85 176T158 471L195 616Q196 629 188 632T149 637H144Q134 637 131

```
637T124 640T121 647Z"></path></g></g><rect
width="1310.4"height="60"x="120"y="220"></rect></g><g data-mml-
node="msup"transform="translate(7183.3, 0)"><g data-mml-node="mi"><path
data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442
430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427
299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350
336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469
151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263
-11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142
143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168
26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217
404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z"></path>
</g><g data-mml-node="mn"transform="translate(572, 413) scale(0.707)"><path
data-c="32"d="M109 429Q82 429 66 447T50 491Q50 562 103 614T235 666Q326 666
387 610T449 465Q449 422 429 383T381 315T301 241Q265 210 201 149L142 93L218
92Q375 92 385 97Q392 99 409 186V189H449V186Q448 183 436 95T421
3V0H50V19V31Q50 38 56 46T86 81Q115 113 136 137Q145 147 170 174T204 211T233
244T261 278T284 308T305 340T320 369T333 401T340 431T343 464Q343 527 309
573T212 619Q179 619 154 602T119 569T109 550Q109 549 114 549Q132 549 151
535T170 489Q170 464 154 447T109 429Z"></path></g></g><g data-mml-
node="mo"transform="translate(8381, 0)"><path data-c="2212"d="M84 237T84
250T98 270H679Q694 262 694 250T679 230H98Q84 237 84 250Z"></path></g><g
data-mml-node="mfrac"transform="translate(9381.3, 0)"><g data-mml-
node="msub"transform="translate(223, 755)"><g data-mml-node="mi"><path data-
c="68"d="M137 683Q138 683 209 688T282 694Q294 694 294 685Q294 674 258
534Q220 386 220 383Q220 381 227 388Q288 442 357 442Q411 442 444 415T478
336Q478 285 440 178T402 50Q403 36 407 31T422 26Q450 26 474 56T513 138Q516
149 519 151T535 153Q555 153 555 145Q555 144 551 130Q535 71 500 33Q466 -10
419 -10H414Q367 -10 346 17T325 74Q325 90 361 192T398 345Q398 404 354
404H349Q266 404 205 306L198 293L164 158Q132 28 127 16Q114 -11 83 -11Q69 -11
59 -2T48 16Q48 30 121 320L195 616Q195 629 188 632T149 637H128Q122 643 122
645T124 664Q129 683 137 683Z"></path></g><g data-mml-
node="mi"transform="translate(576, -150) scale(0.707)"><path data-
c="66"d="M118 -162Q120 -162 124 -164T135 -167T147 -168Q160 -168 171 -155T187
-126Q197 -99 221 27T267 267T289 382V385H242Q195 385 192 387Q188 390 188
397L195 425Q197 430 203 430T250 431Q298 431 298 432Q298 434 307 482T319
540Q356 705 465 705Q502 703 526 683T550 630Q550 594 529 578T487 561Q443 561
443 603Q443 622 454 636T478 657L487 662Q471 668 457 668Q445 668 434 658T419
630Q412 601 403 552T387 469T380 433Q380 431 435 431Q480 431 487 430T498
424Q499 420 496 407T491 391Q489 386 482 386T428 385H372L349 263Q301 15 282
-47Q255 -132 212 -173Q175 -205 139 -205Q107 -205 81 -186T55 -132Q55 -95 76
-78T118 -61Q162 -61 162 -103Q162 -122 151 -136T127 -157L118 -162Z"></path>
</g></g><g data-mml-node="mrow"transform="translate(220, -686)"><g data-mml-
node="mn"><path data-c="32"d="M109 429Q82 429 66 447T50 491Q50 562 103
614T235 666Q326 666 387 610T449 465Q449 422 429 383T381 315T301 241Q265 210
201 149L142 93L218 92Q375 92 385 97Q392 99 409 186V189H449V186Q448 183 436
95T421 3V0H50V19V31Q50 38 56 46T86 81Q115 113 136 137Q145 147 170 174T204
211T233 244T261 278T284 308T305 340T320 369T333 401T340 431T343 464Q343 527
```

```

309 573T212 619Q179 619 154 602T119 569T109 550Q109 549 114 549Q132 549 151
535T170 489Q170 464 154 447T109 429Z"></path></g><g data-mml-
node="mi"transform="translate(500, 0)"><path data-c="6B"d="M121 647Q121 657
125 670T137 683Q138 683 209 688T282 694Q294 694 294 686Q294 679 244 477Q194
279 194 272Q213 282 223 291Q247 309 292 354T362 415Q402 442 438 442Q468 442
485 423T503 369Q503 344 496 327T477 302T456 291T438 288Q418 288 406 299T394
328Q394 353 410 369T442 390L458 393Q446 405 434 405H430Q398 402 367 380T294
316T228 255Q230 254 243 252T267 246T293 238T320 224T342 206T359 180T365
147Q365 130 360 106T354 66Q354 26 381 26Q429 26 459 145Q461 153 479
153H483Q499 153 499 144Q499 139 496 130Q455 -11 378 -11Q333 -11 305 15T277
90Q277 108 280 121T283 145Q283 167 269 183T234 206T200 217T182 220H180Q168
178 159 139T145 81T136 44T129 20T122 7T111 -2Q98 -11 83 -11Q66 -11 57 -1T48
16Q48 26 85 176T158 471L195 616Q196 629 188 632T149 637H144Q134 637 131
637T124 640T121 647Z"></path></g></g><rect
width="1221"height="60"x="120"y="220"></rect></g><g data-mml-
node="msup"transform="translate(10842.3, 0)"><g data-mml-node="mi"><path
data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442
430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427
299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350
336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469
151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263
-11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142
143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168
26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217
404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z"></path>
</g><g data-mml-node="mn"transform="translate(572, 413) scale(0.707)"><path
data-c="32"d="M109 429Q82 429 66 447T50 491Q50 562 103 614T235 666Q326 666
387 610T449 465Q449 422 429 383T381 315T301 241Q265 210 201 149L142 93L218
92Q375 92 385 97Q392 99 409 186V189H449V186Q448 183 436 95T421
3V0H50V19V31Q50 38 56 46T86 81Q115 113 136 137Q145 147 170 174T204 211T233
244T261 278T284 308T305 340T320 369T333 401T340 431T343 464Q343 527 309
573T212 619Q179 619 154 602T119 569T109 550Q109 549 114 549Q132 549 151
535T170 489Q170 464 154 447T109 429Z"></path></g></g><g data-mml-
node="mo"transform="translate(12040, 0)"><path data-c="2B"d="M56 237T56
250T70 270H369V420L370 570Q380 583 389 583Q402 583 409 568V270H707Q722 262
722 250T707 230H409V-68Q401 -82 391 -82H389H387Q375 -82 369 -68V230H70Q56
237 56 250Z"></path></g><g data-mml-
node="mfrac"transform="translate(13040.3, 0)"><g data-mml-
node="msup"transform="translate(220, 676)"><g data-mml-node="mi"><path data-
c="67"d="M311 43Q296 30 267 15T206 0Q143 0 105 45T66 160Q66 265 143 353T314
442Q361 442 401 394L404 398Q406 401 409 404T418 412T431 419T447 422Q461 422
470 413T480 394Q480 379 423 152T363 -80Q345 -134 286 -169T151 -205Q10 -205
10 -137Q10 -111 28 -91T74 -71Q89 -71 102 -80T116 -111Q116 -121 114 -130T107
-144T99 -154T92 -162L90 -164H91Q101 -167 151 -167Q189 -167 211 -155Q234 -144
254 -122T282 -75Q288 -56 298 -13Q311 35 311 43ZM384 328L380 339Q377 350 375
354T369 368T359 382T346 393T328 402T306 405Q262 405 221 352Q191 313 171
233T151 117Q151 38 213 38Q269 38 323 108L331 118L384 328Z"></path></g><g
data-mml-node="mo"transform="translate(477, 363) scale(0.707)"><g data-

```

```
c="2034"><path data-c="2032"d="M79 43Q73 43 52 49T30 61Q30 68 85 293T146
528Q161 560 198 560Q218 560 240 545T262 501Q262 496 260 486Q259 479 173
263T84 45T79 43Z"></path><path data-c="2032"d="M79 43Q73 43 52 49T30 61Q30
68 85 293T146 528Q161 560 198 560Q218 560 240 545T262 501Q262 496 260
486Q259 479 173 263T84 45T79 43Z"transform="translate(275, 0)"></path><path
data-c="2032"d="M79 43Q73 43 52 49T30 61Q30 68 85 293T146 528Q161 560 198
560Q218 560 240 545T262 501Q262 496 260 486Q259 479 173 263T84 45T79
43Z"transform="translate(550, 0)"></path></g></g></g><g data-mml-
node="mrow"transform="translate(264.7, -686)"><g data-mml-node="mn"><path
data-c="36"d="M42 313Q42 476 123 571T303 666Q372 666 402 630T432 550Q432 525
418 510T379 495Q356 495 341 509T326 548Q326 592 373 601Q351 623 311 626Q240
626 194 566Q147 500 147 364L148 360Q153 366 156 373Q197 433 263 433H267Q313
433 348 414Q372 400 396 374T435 317Q456 268 456 210V192Q456 169 451 149Q440
90 387 34T253 -22Q225 -22 199 -14T143 16T92 75T56 172T42 313ZM257 397Q227
397 205 380T171 335T154 278T148 216Q148 133 160 97T198 39Q222 21 251 21Q302
21 329 59Q342 77 347 104T352 209Q352 289 347 316T329 361Q302 397 257 397Z">
</path></g><g data-mml-node="mi"transform="translate(500, 0)"><path data-
c="6B"d="M121 647Q121 657 125 670T137 683Q138 683 209 688T282 694Q294 694
294 686Q294 679 244 477Q194 279 194 272Q213 282 223 291Q247 309 292 354T362
415Q402 442 438 442Q468 442 485 423T503 369Q503 344 496 327T477 302T456
291T438 288Q418 288 406 299T394 328Q394 353 410 369T442 390L458 393Q446 405
434 405H430Q398 402 367 380T294 316T228 255Q230 254 243 252T267 246T293
238T320 224T342 206T359 180T365 147Q365 130 360 106T354 66Q354 26 381 26Q429
26 459 145Q461 153 479 153H483Q499 153 499 144Q499 139 496 130Q455 -11 378
-11Q333 -11 305 15T277 90Q277 108 280 121T283 145Q283 167 269 183T234
206T200 217T182 220H180Q168 178 159 139T145 81T136 44T129 20T122 7T111 -2Q98
-11 83 -11Q66 -11 57 -1T48 16Q48 26 85 176T158 471L195 616Q196 629 188
632T149 637H144Q134 637 131 637T124 640T121 647Z"></path></g></g><rect
width="1310.4"height="60"x="120"y="220"></rect></g><g data-mml-
node="msup"transform="translate(14590.6, 0)"><g data-mml-node="mi"><path
data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442
430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427
299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350
336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469
151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263
-11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142
143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168
26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217
404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z"></path>
</g><g data-mml-node="mn"transform="translate(572, 413) scale(0.707)"><path
data-c="33"d="M127 463Q100 463 85 480T69 524Q69 579 117 622T233 665Q268 665
277 664Q351 652 390 611T430 522Q430 470 396 421T302 350L299 348Q299 347 308
345T337 336T375 315Q457 262 457 175Q457 96 395 37T238 -22Q158 -22 100 21T42
130Q42 158 60 175T105 193Q133 193 151 175T169 130Q169 119 166 110T159 94T148
82T136 74T126 70T118 67L114 66Q165 21 238 21Q293 21 321 74Q338 107 338
175V195Q338 290 274 322Q259 328 213 329L171 330L168 332Q166 335 166 348Q166
366 174 366Q202 366 232 371Q266 376 294 413T322 525V533Q322 590 287 612Q265
626 240 626Q208 626 181 615T143 592T132 580H135Q138 579 143 578T153 573T165
```

```

566T175 555T183 540T186 520Q186 498 172 481T127 463Z"></path></g></g><g
data-mml-node="mo"transform="translate(15788.4, 0)"><path data-
c="2212"d="M84 237T84 250T98 270H679Q694 262 694 250T679 230H98Q84 237 84
250Z"></path></g><g data-mml-node="mfrac"transform="translate(16788.6, 0)">
<g data-mml-node="msub"transform="translate(223, 755)"><g data-mml-
node="mi"><path data-c="68"d="M137 683Q138 683 209 688T282 694Q294 694 294
685Q294 674 258 534Q220 386 220 383Q220 381 227 388Q288 442 357 442Q411 442
444 415T478 336Q478 285 440 178T402 50Q403 36 407 31T422 26Q450 26 474
56T513 138Q516 149 519 151T535 153Q555 153 555 145Q555 144 551 130Q535 71
500 33Q466 -10 419 -10H414Q367 -10 346 17T325 74Q325 90 361 192T398 345Q398
404 354 404H349Q266 404 205 306L198 293L164 158Q132 28 127 16Q114 -11 83
-11Q69 -11 59 -2T48 16Q48 30 121 320L195 616Q195 629 188 632T149 637H128Q122
643 122 645T124 664Q129 683 137 683Z"></path></g><g data-mml-
node="mi"transform="translate(576, -150) scale(0.707)"><path data-
c="66"d="M118 -162Q120 -162 124 -164T135 -167T147 -168Q160 -168 171 -155T187
-126Q197 -99 221 27T267 267T289 382V385H242Q195 385 192 387Q188 390 188
397L195 425Q197 430 203 430T250 431Q298 431 298 432Q298 434 307 482T319
540Q356 705 465 705Q502 703 526 683T550 630Q550 594 529 578T487 561Q443 561
443 603Q443 622 454 636T478 657L487 662Q471 668 457 668Q445 668 434 658T419
630Q412 601 403 552T387 469T380 433Q380 431 435 431Q480 431 487 430T498
424Q499 420 496 407T491 391Q489 386 482 386T428 385H372L349 263Q301 15 282
-47Q255 -132 212 -173Q175 -205 139 -205Q107 -205 81 -186T55 -132Q55 -95 76
-78T118 -61Q162 -61 162 -103Q162 -122 151 -136T127 -157L118 -162Z"></path>
</g></g><g data-mml-node="mrow"transform="translate(220, -686)"><g data-mml-
node="mn"><path data-c="36"d="M42 313Q42 476 123 571T303 666Q372 666 402
630T432 550Q432 525 418 510T379 495Q356 495 341 509T326 548Q326 592 373
601Q351 623 311 626Q240 626 194 566Q147 500 147 364L148 360Q153 366 156
373Q197 433 263 433H267Q313 433 348 414Q372 400 396 374T435 317Q456 268 456
210V192Q456 169 451 149Q440 90 387 34T253 -22Q225 -22 199 -14T143 16T92
75T56 172T42 313ZM257 397Q227 397 205 380T171 335T154 278T148 216Q148 133
160 97T198 39Q222 21 251 21Q302 21 329 59Q342 77 347 104T352 209Q352 289 347
316T329 361Q302 397 257 397Z"></path></g><g data-mml-
node="mi"transform="translate(500, 0)"><path data-c="6B"d="M121 647Q121 657
125 670T137 683Q138 683 209 688T282 694Q294 694 294 686Q294 679 244 477Q194
279 194 272Q213 282 223 291Q247 309 292 354T362 415Q402 442 438 442Q468 442
485 423T503 369Q503 344 496 327T477 302T456 291T438 288Q418 288 406 299T394
328Q394 353 410 369T442 390L458 393Q446 405 434 405H430Q398 402 367 380T294
316T228 255Q230 254 243 252T267 246T293 238T320 224T342 206T359 180T365
147Q365 130 360 106T354 66Q354 26 381 26Q429 26 459 145Q461 153 479
153H483Q499 153 499 144Q499 139 496 130Q455 -11 378 -11Q333 -11 305 15T277
90Q277 108 280 121T283 145Q283 167 269 183T234 206T200 217T182 220H180Q168
178 159 139T145 81T136 44T129 20T122 7T111 -2Q98 -11 83 -11Q66 -11 57 -1T48
16Q48 26 85 176T158 471L195 616Q196 629 188 632T149 637H144Q134 637 131
637T124 640T121 647Z"></path></g></g><rect
width="1221"height="60"x="120"y="220"></rect></g><g data-mml-
node="msup"transform="translate(18249.6, 0)"><g data-mml-node="mi"><path
data-c="78"d="M52 289Q59 331 106 386T222 442Q257 442 286 424T329 379Q371 442
430 442Q467 442 494 420T522 361Q522 332 508 314T481 292T458 288Q439 288 427

```

```

299T415 328Q415 374 465 391Q454 404 425 404Q412 404 406 402Q368 386 350
336Q290 115 290 78Q290 50 306 38T341 26Q378 26 414 59T463 140Q466 150 469
151T485 153H489Q504 153 504 145Q504 144 502 134Q486 77 440 33T333 -11Q263
-11 227 52Q186 -10 133 -10H127Q78 -10 57 16T35 71Q35 103 54 123T99 143Q142
143 142 101Q142 81 130 66T107 46T94 41L91 40Q91 39 97 36T113 29T132 26Q168
26 194 71Q203 87 217 139T245 247T261 313Q266 340 266 352Q266 380 251 392T217
404Q177 404 142 372T93 290Q91 281 88 280T72 278H58Q52 284 52 289Z"></path>
</g><g data-mml-node="mn"transform="translate(572, 413) scale(0.707)"><path
data-c="33"d="M127 463Q100 463 85 480T69 524Q69 579 117 622T233 665Q268 665
277 664Q351 652 390 611T430 522Q430 470 396 421T302 350L299 348Q299 347 308
345T337 336T375 315Q457 262 457 175Q457 96 395 37T238 -22Q158 -22 100 21T42
130Q42 158 60 175T105 193Q133 193 151 175T169 130Q169 119 166 110T159 94T148
82T136 74T126 70T118 67L114 66Q165 21 238 21Q293 21 321 74Q338 107 338
175V195Q338 290 274 322Q259 328 213 329L171 330L168 332Q166 335 166 348Q166
366 174 366Q202 366 232 371Q266 376 294 413T322 525V533Q322 590 287 612Q265
626 240 626Q208 626 181 615T143 592T132 580H135Q138 579 143 578T153 573T165
566T175 555T183 540T186 520Q186 498 172 481T127 463Z"></path></g></g></g>
</g></svg></mjx-container></span></div><div>Numerical Solution:<br>The
numerical solution involves discretizing the wall into nodes and applying
the energy balance at each node to solve for the temperature distribution.
By solving the system of equations formed from the energy balance at each
node, we can obtain the numerical temperature distribution.</div><div>To
compare the two solutions, we can plot the temperature distribution obtained
from the analytical solution against the temperature distribution obtained
from the numerical solution. This comparison will allow us to assess the
accuracy of the numerical method in approximating the temperature profile in
the plane wall.</div><div>By overlaying the analytical solution curve onto
the numerical solution curve, we can visually inspect how closely the
numerical method aligns with the analytical solution. Any deviations between
the two curves can provide insights into the accuracy and reliability of the
numerical approach in solving the heat transfer problem in the plane wall.
</div><div>This comparison will help us evaluate the effectiveness of the
numerical method in capturing the temperature distribution and validate its
results against the analytical solution.</div><h1 id="10.-video-
presentation-preparation">10. Video Presentation Preparation</h1><div>For
the Video Presentation Preparation, the student should focus on creating a
concise and engaging presentation that effectively communicates the key
aspects of their solution. Here are some tips for preparing the video:</div>
<ol><li><div>Introduction:</div><ul><li>Start the video by introducing
yourself and providing a brief overview of the problem statement.</li>
<li>Clearly state the objectives of the analysis and what the audience can
expect from the presentation.</li></ul></li><li><div>Solution Overview:
</div><ul><li>Summarize the analytical and numerical methods used to solve
the heat transfer problem.</li><li>Highlight the key steps involved in
deriving the temperature distribution and solving for the maximum
temperatures.</li></ul></li><li><div>MATLAB Implementation:</div><ul>
<li>Demonstrate how the numerical solution was implemented in MATLAB,
including the setup of nodes, energy balance equations, and temperature

```


calculation.

- Show snippets of the code and explain the logic behind each step.

Results and Analysis:

- Present the temperature distribution **plot** and discuss any trends or observations.
- Analyze the variation of maximum temperatures with the number of nodes and the location of maximum temperatures.

Comparison with Analytical Solution:

- Overlay the analytical solution on the numerical results **plot** and discuss the agreement or discrepancies between the two.

Conclusion:

- Summarize the main findings of the analysis and emphasize the significance of the results.
- Conclude with any insights gained from the study and potential areas **for** further investigation.

Presentation Tips:

- Practice the presentation multiple times to ensure clarity and coherence.
- Use visual aids such as slides or diagrams to enhance understanding.
- Maintain a confident and engaging demeanor throughout the video.

By following these guidelines, the student can create a compelling video presentation that effectively communicates their solution to the heat transfer problem.

11. Conclusion

In conclusion, the analysis of the 1D plane wall heat transfer problem using both analytical and numerical **methods** provides valuable insights into the temperature distribution and behavior within the system. By deriving the temperature distribution equation and implementing a numerical solution approach, we were able to visualize how heat is transferred through the wall and identify key parameters such as maximum temperatures and their locations.

The comparison between the analytical and numerical solutions showcased the effectiveness of numerical **methods** in solving **complex** heat transfer problems with practical boundary conditions. The agreement between the two solutions validates the accuracy of the numerical approach and its applicability in **real-world** engineering scenarios.

Overall, this homework assignment has deepened our understanding of heat transfer principles, numerical **methods**, and the importance of considering various boundary conditions in analyzing thermal systems. The combination of analytical and numerical techniques has provided a comprehensive solution to the given problem, highlighting the significance of both approaches in engineering analysis and design.